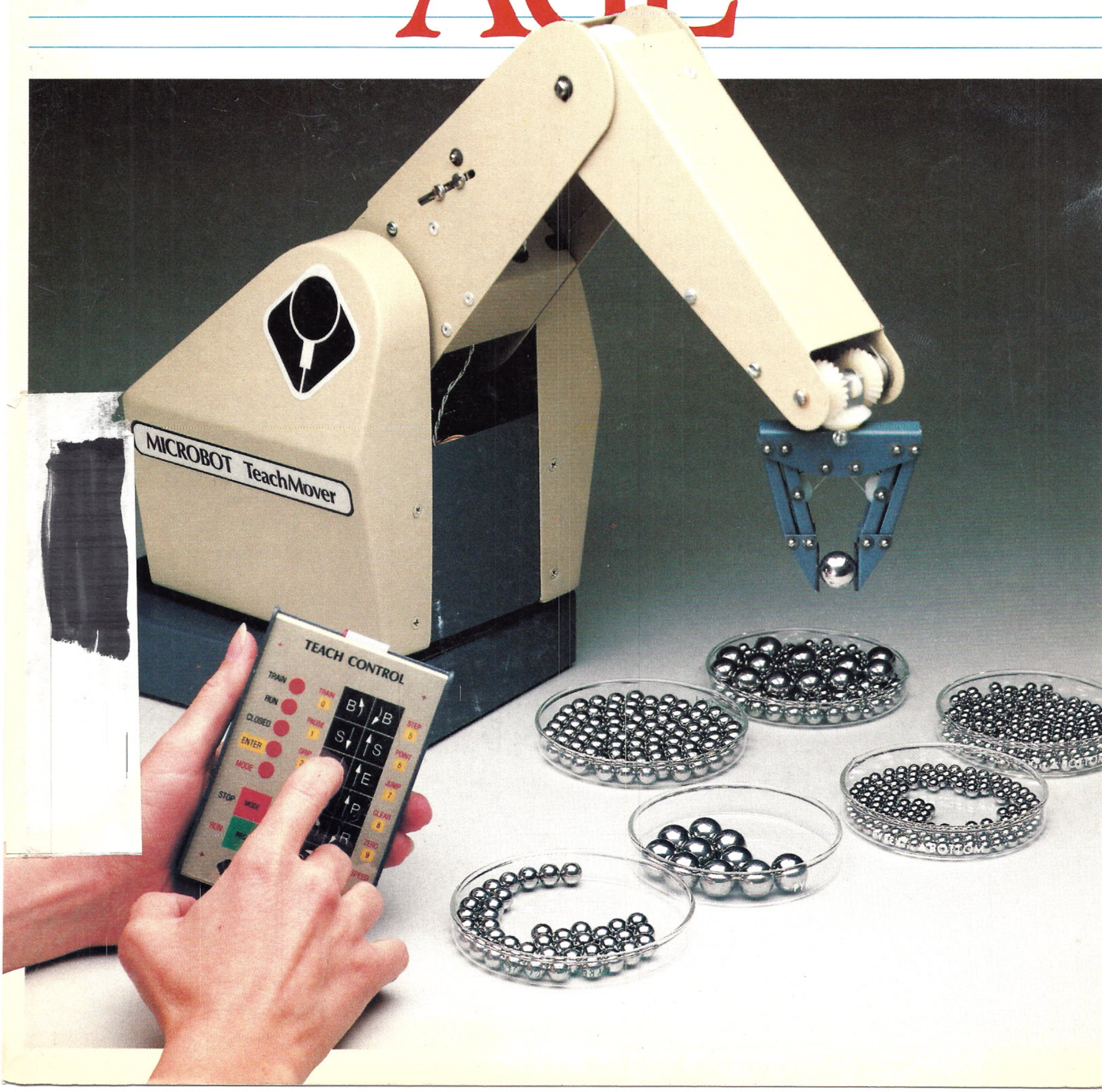
JULY/AUGUST 1982 VOL. 4 NO. 4

$3.00 USA/$3.50 CANADA

THE JOURNAL OF INTELLIGENT MACHINES

ROBOTICS AGE™



CIRCLE 17

MS/PhD Electrical, Mechanical, Computer Engineering, Computer Science

Western Electric's Engineering Research Center in Princeton, N.J. has R&D opportunities in two research groups: Advanced Automation and Test Systems.

Here is a rare opportunity to be involved with problems in their entirety, from the research phase through implementation in the manufacturing environment.

Advanced Automation—EE, Computer Engineering

Will be a Member of Research Staff in Advanced Automation Studies Group developing flexible, reprogrammable automation technology for manufacturing operations. Objectives include rapid introduction of new codes, smaller volumes of each code, shorter manufacturing life cycles. Activities include robotics, machine vision systems, control systems, special automation projects. Your credentials should demonstrate advanced degree and good background in Electrical or Computer Engineering—with particular emphasis on microprocessor control systems.

Advanced Automation—ME

Will be a Member of Research Staff in Advanced Automation Studies Group, and will work closely in tandem with the Engineer described above. Together, you'll be taking projects phase-by-phase from research right through manufacturing implementation. In addition to advanced degree in Mechanical Engineering, your credentials should demonstrate strong background in design, and understanding of associated electronics and microprocessor control systems.

Test System Studies—EE, Computer Science

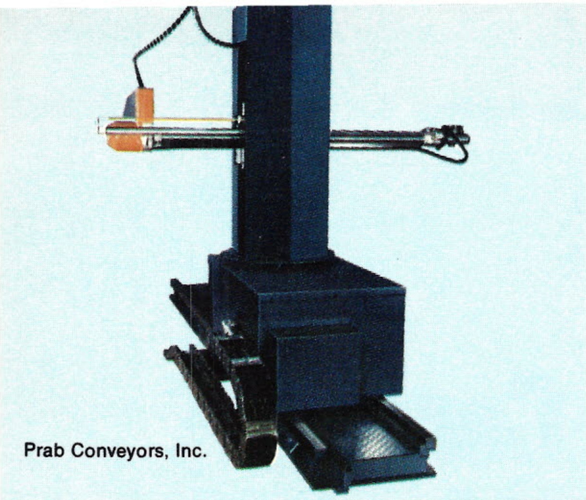
Will be involved in establishing new techniques for testing VLSI-based components, subsystems and systems. Your work in testability, test generation, diagnostics and fault tolerant computing techniques will reach back into the basic design function. In addition to advanced degree in EE and/or Computer Science, we require experience in one or more of the following:

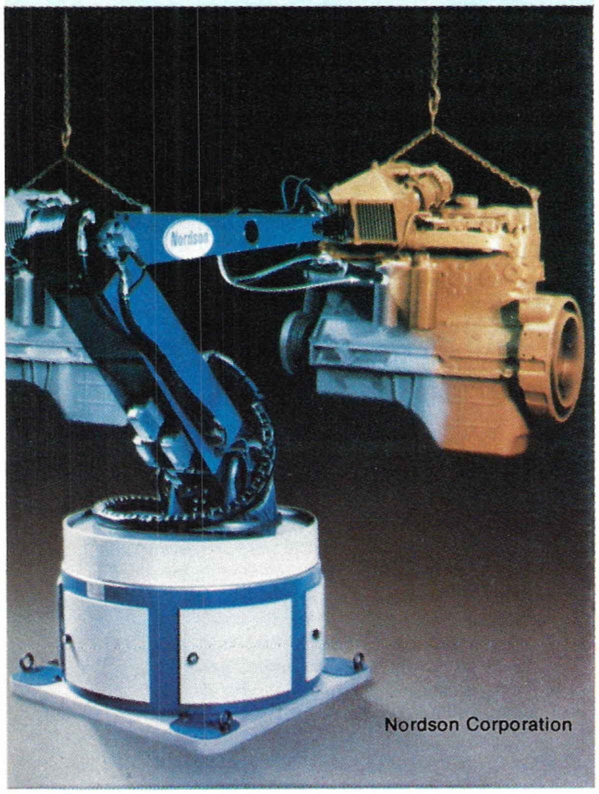
- **Automatic Test Generation for analog, digital or hybrid circuits**
- **Simulation from Architectural level to logic level**
- **Theoretical aspects of microcode and microcoded systems with emphasis on reliability, testability and CAD tools**
- **Logic synthesis and related CAD tool development, algorithm design and evaluation of Artificial intelligence especially as it might apply to aiding a designer in the development and evaluation of design alternatives**
- **Structured design techniques such as Level Sensitive Scan Design and tools supporting such methodologies**
- **Built-in-test and self-test techniques for microprocessor-based products**
- **In-circuit testing of state-of-the-art product**
- **System-level test and high resolution system diagnostics**

For prompt, confidential consideration... and briefing on excellent salaries, benefits (including relocation assistance), and career potential... please send resume with salary history to: Mr. N. E. Leaver, Ref. 101, Western Electric, Engineering Research Center, P.O. Box 900, Princeton, NJ 08540. Please identify position of interest.

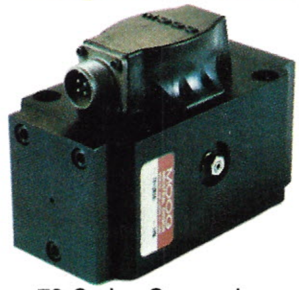
An equal opportunity employer

Western Electric



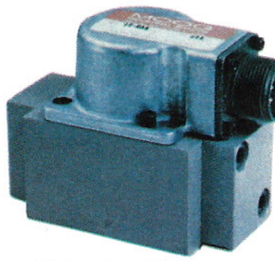


Moog Servovalves

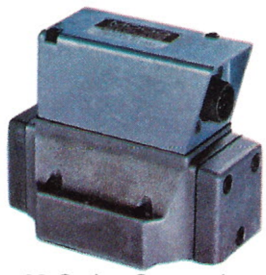


78 Series Servovalve

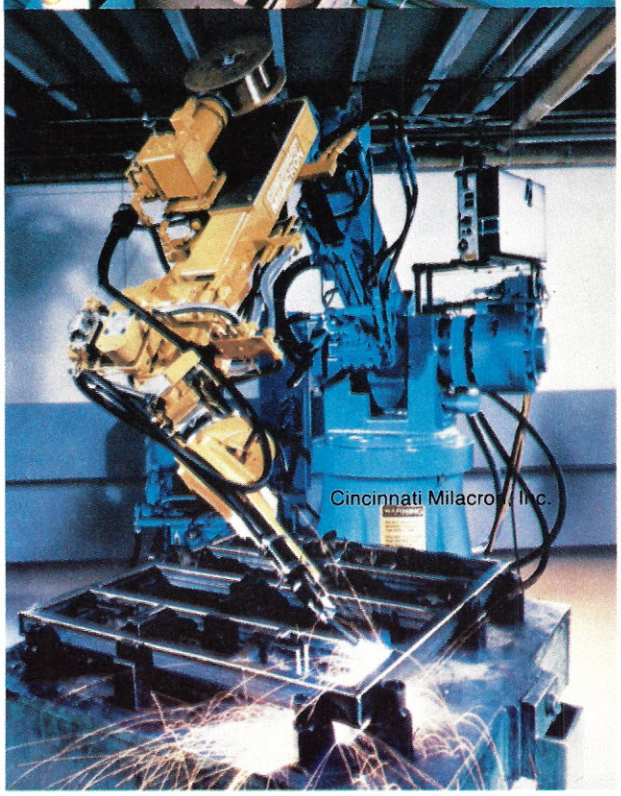
760 Series Servovalve



77 Series Servovalve



62 Series Servovalve



Industrial robots depend upon Moog servovalves for control of position, velocity, acceleration and deceleration

The robot population is growing exponentially as manufacturers seek higher productivity and lower operating costs.

A high percentage of the industrial robot population will be controlled by Moog Servovalves. These valves provide precise control of the many functions on today's robots, which are performing a wide range of tasks, including materials handling, all types of welding, painting, and intricate assembly operations. Moog provides a full range of servovalve sizes to fit the needs of the complete spectrum of robots. These valves are available with intrinsically safe coils for painting robots and others designed for hazardous environments.

To assure robot users of production continuity, we have fully equipped and expertly staffed servovalve service and repair departments in East Aurora, NY; Monterey Park, CA; England, France, West Germany, Sweden, Italy, Japan, Australia and Brazil.

For more information on Moog Servovalves, our "fast turnaround" repair service, or factory-rebuilt "remanufactured" valves, write to Moog Inc. Industrial Division, East Aurora, NY 14052. Or telephone us at 716/652-2000.

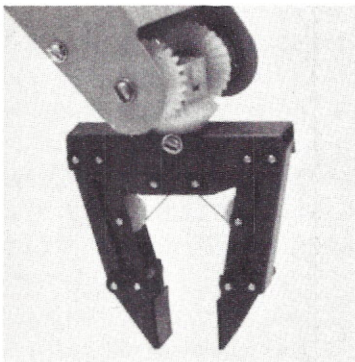
MOOG
INDUSTRIAL DIVISION

CIRCLE 7

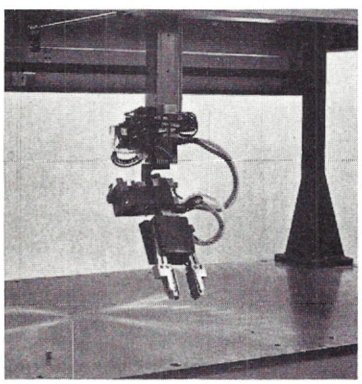
THE JOURNAL OF INTELLIGENT MACHINES

ROBOTICS AGE™

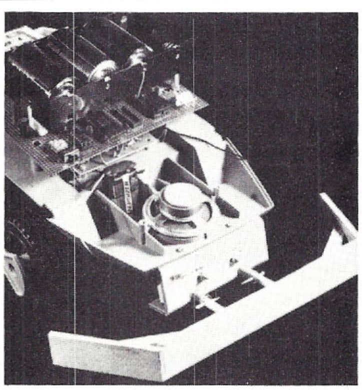
JULY/AUGUST 1982 VOL. 4 NO. 4



12



29



41

EDITORIAL

FEATURES

DEPARTMENTS

About the Cover: On this issue's cover we find a photographic illustration of an idea: sorting small parts with a manipulator. This staged photograph is not quite complete. The Microbot TeachMover illustrates the manipulator component of an application that would require sensor inputs such as vision, and a computer to control the whole show.

The photograph for the cover is by Paul Avis. The balls and petri dishes are supplied courtesy of precision ball bearing manufacturer M.P.B. Inc., Keene, N.H. Thanks also to Dick Switzer of Microbot for loaning us the TeachMover.

Editorial

A Telecommunications Robot

BY CARL HELMERS

We tend to stereotype our conceptions. The meaning of a cherished concept sharply limits the set of possible instances. Such stereotypes can obscure important parts of a much larger truth. One common oversimplification is the requirement that a robot have actual mechanical parts, typically of anthropomorphic shape. This is the prejudice of the science fiction robot concept, an important source of inspiration for the experimental engineering of robots. The equation of robots with mechanical arms in processes involving physical activities is responsible for the common image of the robotic assembly arm in a factory. Anthropomorphism is the premise behind the success of show robots.

The larger framework in which we operate as people who design and apply intelligent machines involves far more than just mechanical action. The human form is a good start for modeling the mechanically active robot, but there is more to automatic control and planning than mere physical actions. Robotics engineering certainly can involve a machine system which in fact has no mechanical parts, just electrical and electronic parts. Such robots exist in a logical world of mathematical forms, with a more indirect connection to the real world of human activities.

One example of such a "logical robot" concept is the telecommunications robot. This is an electronic device, a computer, hooked up to a telecommunications system provided by the phone companies. The permissible sets of operations are extensions of the telecommunications robot's presence through the address space of the phone network. Telecommunications robots in their simplest forms are one of the most common consumer telephone accessories. We see advertisements for many different brands and variations on the "autodialer" or "memory dialer" theme. In more elaborate, but still relatively simple form, the subject of this writing is a commercially oriented communications robot we're actually implementing for *Robotics Age*.

The extension of the telecommunications robot's sensing and effecting presence can be active or passive. An active telecommunications robot is one which actually places calls, then communicates with another telecommunications robot or with a person. A passive telecommunications robot is one which merely answers calls from people or other telecommunications robots. In either case, the telecommunications machine fits our criteria for being called a robot:

Editor
Carl T. Helmers, Jr.

Associate Editor
Raymond G. A. Cote

Editorial Assistant
Perry B. Anable

Technical Associates
Walter R. Banks
Scott Nowell

Typography
Reprographics

Production/Design
Nancy Estle

Consulting Designer
Robert Tinney

Publisher
Glen W. Swanson

Advertising & Marketing
Laura A. Hanson

Business Systems, Circulation & Dealer Sales
James E. Bingham
Gordon Peery

Printing
Wellesley Press

Advertising Sales Offices

General Advertising Sales
Laura A. Hanson
Robotics Age Inc.
174 Concord Street
Peterborough, NH 03458
603-924-7136

Western Sales
Jules E. Thompson, Inc.
1290 Howard Ave., Suite 303
Burlingame, CA 94010
415-348-8222
In Texas call 713-731-2605

Southern California
Jules E. Thompson, Inc.
2560 Via Tejon
Palos Verdes Estates, CA 90274
213-378-8361

Mid-Atlantic, Northeast
CEL Associates, Inc.
27 Adams Street
Braintree, MA 02184
617-848-9306

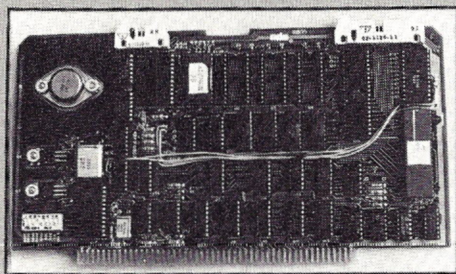
ROBOTICS AGE — USPS 523850 (ISSN 0197-1905) is published six times a year in January, March, May, July, September and November by Robotics Age Inc., Strand Building, 174 Concord Street, Peterborough, N.H. 03458, phone (603) 924-7136. Address subscriptions, change of address, USPS Form 3579, and fulfillment questions to *Robotics Age* SUBSCRIPTIONS, P.O. Box 358, Peterborough, N.H. 03458. Second class postage paid at Peterborough, N.H. and at additional mailing offices.

Subscriptions are $15 for one year (6 issues), $28 for two years (12 issues), $39 for three years (18 issues) in the USA and its possessions. In Canada and Mexico, subscriptions are $17 for one year, $32 for two years, $45 for three years. For other countries, subscriptions are $19 for one year surface delivery. Air delivery to selected areas at additional charges, rates upon request. Single copy price is $3 in the U.S., $3.50 in Canada and Mexico, $4 in Europe, and $4.50 elsewhere. Foreign subscriptions and single copy sales should be remitted in United States funds drawn on a U.S. bank.

Address all editorial correspondence to the Editor, at *Robotics Age*, Strand Building, 174 Concord Street, Peterborough, N.H. 03458. Opinions expressed by the authors of articles are not necessarily those of *Robotics Age*. To aid in preparation of acceptable articles, the *Robotics Age* Authors' Guide is available upon request if accompanied by a self-addressed 8½ by 11 inch envelope with first class postage for 3 ounces. Unacceptable manuscripts will be returned if accompanied by a self-addressed envelope with sufficient first class postage. Not responsible for lost manuscripts or photos.

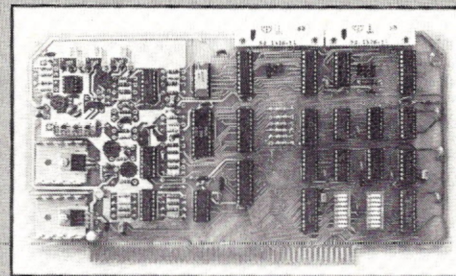
S-100 UPDATE SHEET #3

the equation:



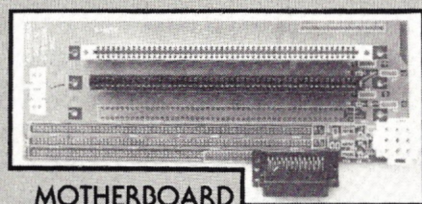
INTELLIGENCE

+



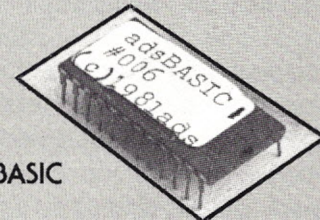
SPEECH SYNTHESIS

+



MOTHERBOARD

+



BASIC

= **Robotics Applications, Artificial Intelligence, Machine Control, & More!**

the solution:

= Ackerman Digital Systems, Inc.

Our Single Board Computer, utilizing the powerful MC6809 microprocessor, gives you exceptional computing power. Speech synthesis with the Votrax SC-01 * I.C. for unlimited capabilities. Our powerful small Basic, (in 2716 eprom), residing on-board in our MC6809 S.B.C. gets your system up and running at a low cost, but with plenty of power. All of the above combined into our 3 slot Mother Board means portable power!

*VOTRAX. TRADEMARK OF FEDERAL SCREW WORKS, INC.

We have it all!

And a lot more, including a prom programmer S-100 card, (programs 14 different devices), sound effects, a prototyping card, all compatable with the I.E.E.E. 696/S-100 bus.

You need to know about us!

As the saying goes, we have solutions looking for a problem, well-engineered solutions. Call or write us today for a complete catalog.

ads

Ackerman Digital Systems, Inc.
110 N. York Road; Suite 208
Elmhurst, Illinois 60126
(312)530-8992

CIRCLE 1

- It has software running with knowledge of real time in a computing device.
- This software reacts to and controls sensor inputs from the external environment.
- This software controls effectors that interact with its external environment of people, other systems, or physical objects.
- The intelligent machine system has a purpose, which dominates the design choices.

The telecommunications robot has no physical arms or legs. The closest thing to a mechanical part of its operation is the possible activation of any relays still installed in the phone network.

The software intelligence of the telecommunications robot is provided by a real-time program which monitors and controls the activity on a phone line. The telecommunications robot exists in an environment of the telephone network, supervised by the people who need this kind of service. In one sense, the whole phone network can be viewed as a robotic universe within which other telecommunications robots can communicate. Any centralized computer with time-sharing access is an example of such a telecommunications robot device. A personal computer set up with a modem and software that can automatically send and receive electronic mail is within this universe of telecommunications robots.

The telecommunications universe requires specific engineering and design considerations that are non-trivial. It is certainly a real-time universe. Events occur at various times. Services are provided by the phone network on a contention basis.

The telecommunications universe itself is built upon practical solutions to problems of queuing theory and information theory. The real-time nature of the telecommunications network viewed as a robot, as well as its queuing theory considerations, can be impressively demonstrated by the impact of a recent natural event upon a portion of the network.

This spring, there was a mild earthquake near Concord, New Hampshire. The quake was felt throughout the state. At our home, my wife and I felt the grating, grinding vibration, and wondered what it was.

All sorts of thoughts passed through our heads. Did the furnace blow up? A check of the basement disproved that idea. Did the two feet of snow on our roof come crashing to the ground? A glance outside showed no change under the eaves. Did a logging truck full of pine trees come and park in the driveway? After treating and discarding all these trial hypotheses in turn, we turned to the phone network.

I called our next door neighbor. I said something to the effect of "Chip, did you feel that?" He replied with words like "Yes, did you feel it too?" We both immediately said, "earthquake." I repeated this conversation at least twice in the next few minutes with friends in the next town, then with a relative across the state. Each phone call became more difficult to accomplish, with several mis-dials and hung lines. Imagine my calls being repeated across the geographical region that had felt the earthquake.

The simultaneous signal of the earthquake transformed itself into a statistical aberration in the phone network, as a great proportion of the inhabitants of the state dialed their friends and relations to hold the same sort of conversation. The earthquake violated a statistical assumption in the queuing theory analysis of the phone network's design — the assumption that calls occur at random within the geographical area served by the network. The earthquake's simultaneous instigation of calls by a significant portion of the subscriber base caused a temporary service breakdown of the phone network, viewed as a robot serving its subscribers. The computing function which is network switching was overloaded by the subscriber requests.

The statistical assumption problem here was obvious. More subtle and less obvious statistical pitfalls exist in the design of real-time control systems. The dynamics of the interaction of a subscriber's telecommunications robot with the phone network involve solving problems of dealing with the normal real-time responses and properties of the network. These are the underlying challenges of the telecommunications robot design problem.

The telecommunications robot, to be successful, must deal with exception conditions as they occur, must exist within an appropriate (but always limited) local hardware environment, must check its own integrity from time to time, and must keep its files in a way that allows recovery from the exception conditions when they occur. It senses messages sent in over the phone network. Its responses to these messages and questions define its purpose and function in the real world. In addition to network contingencies, if it is dealing with a multitude of individual interacting robots or people, it must allow for another level of exceptions. The result is a robot which is largely a software design, with a small proportion of hardware. The telecommunications robot emphasizes the importance of software to robotic system design.

Editorial

A Robot Must Perform Good Works. Whether the goal is amusement and experimental design fun, or a more serious practical purpose, the robot must do good works. Good is evaluated in a human context. A work is whatever the robot does. In the specific case of our telecommunications robot design for *Robotics Age*, the "good works" of the design are several goals accomplished through the vehicle of standard Bell 103 modem communications techniques:

- Robotics News Service
- Subscription service: new subscriptions, renewal, and address change
- Classified advertising placement
- Article manuscript transmission in machine-readable form.

The Robotics News Service is a way of electronically publishing short tidbits of what is happening in the world of robotics as seen from the *Robotics Age* editorial desk. The Robotics News Service function of the Robotics Information Service is simply a way of making available a text file that can be inspected at will by the person placing a call to our telecommunications robot. This news distribution function is a characteristic of any "bulletin board" system. It provides a means of disseminating quick and timely information to the people who use this telecommunications robot.

In the Subscription Service functions, a subscriber with a modem can place a call to enter various off-line requests for service from our fulfillment department. The implemented services are: adding a new subscription, changing address, or extending a subscription. The notes taken by the telecommunications robot at present are batched for processing by our circulation list system. In cases of new subscriptions and extensions, payment for services can be accomplished by a sequence to enter and verify credit card information.

The Classified Advertisement functions of the Robotics Information Service provide simple text editing, exact calculation of the advertising cost for the advertisement, and verification of the copy used in the advertisement. Once the advertisement is verified and accepted by the person placing the call, the final part of the interaction is entry of credit card data. The direct entry of classified advertisements gives the advertiser immediate confirmation of our interpretation of his keystrokes. This eliminates one additional stage of keystroking when we send the information into a typesetting system via communications links.

A similar quality improvement is possible by eliminating a keystroking step in our article submissions. Perhaps the most inherently error prone operation is keystroking — taking a manuscript and typing it into the photocomposition machine. In the production of a magazine like ours, this manual step is followed by a manual step of proofreading and correction. The process is fraught with opportunities to miss typographical errors and introduce strange transformations of meaning.

The reliability of the process can be improved if we use machine-readable copy obtained from the author. One way this copy can be obtained is through telecommunications. Here, the author is assigned an article number, which becomes the account number that is used to verify authorized access to the text-recording system. (It should be noted that use of compatible magnetic media and a physical delivery service such as Federal Express or UPS can also serve this purpose and is probably less expensive.)

Editorial

The Robotics Age Information Service System. To achieve our goals, these functions are realized in the form of a software robot that can run automatically on a 24-hour basis in a small computer tied to the phone network. As a robot, the sensed inputs are modem tones from the caller and the sensing of the phone's ringing. Its effector outputs are modem tones, and electronic means to "pick up" or "hang up" the telephone. Miscellaneous logging activities are performed on the mass storage media of the computer and on a hard-copy transcript for quick manual verification. And, as noted above, this robot — an automatic untended system — has a purpose defined within the context of a magazine publishing organization. Different purposes would apply in cases such as modem-order retailing from catalogs, extended news service ideas, and so on.

The diagram of Figure 1 shows the present concept of this system as it is being implemented. The design is a joint effort of myself and my associate Ray Cote. Ray implemented the system for one of our Apple II computers using the Apple Pascal (UCSD) system. Our modem function is provided by the Hayes Microcomputer "Smartmodem" product, running through a CCS 7710 serial port to the Apple II. Since the programs are written in a machine-independent form using a machine-independent system, we can also run the same software on any Pascal-compatible computer. (We have demonstrated similar programs running without change on PDP-11 and Western Digital equipment.)

Since there is an ever present possibility of error, we track all activities of the system with a hard-copy log. The actual hardware we employ is an IDS-460. In present versions of the system, we do not have clock/calendar hardware. Thus, logging operations record only the sequence of interactions, without time of day, although the date is set manually once a day.

Where Is it now? The Robotics Information Service, as of this writing in mid-May, is at a stage of final testing. The equipment and programs work in a local mode run with our own terminals through the local end of the phone network. In order to phase the system into total operation, we are prepared to make tests on a scheduled basis with readers of *Robotics Age.* We solicit our readers' help in testing out the system, trying to "break" the design as it were.

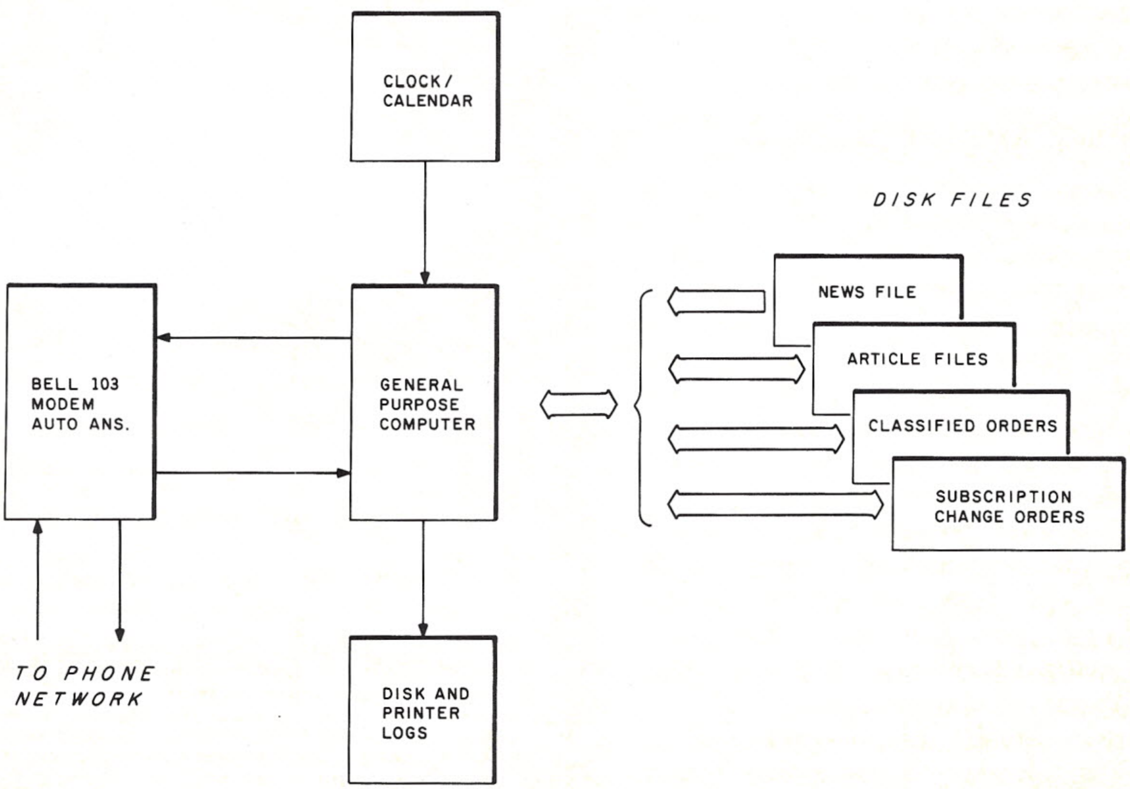


Figure 1: The Robotics Information Service System. This system diagram shows the general idea of the new Robotics Information Service which we are implementing for *Robotics Age* subscribers. The system is a special purpose software control robot operating in an environment of a small computer connected to the phone network with a readily available intelligent modem. Its roots and inspiration in past systems include a variety of community bulletin board projects, time-sharing software techniques, and similar commonly used practices of the computing world.

Listing 1 illustrates a typical dialog which is produced by the system as of this writing. The listing was produced as a system log recorded on disk. For quality of magazine reproduction, we printed these examples on a Diablo printer, although our usual hard-copy logging device is the dot matrix IDS-460.

If you want to participate in the tests and give us some feedback on the user interface, write us or give a call at (603) 924-7136. We'll reply with a scheduled time and a phone number you can use to dial the service from your Bell-103 originate style modem. Then, in a "let's pretend" mode, you can try out the interface as if you were really using it. Your feedback on the design of the software will be much appreciated after you participate in this way. We'll keep readers updated on the results of these tests in future editorial writings. After the system is working acceptably, we'll add a phone line and keep it in operation on a continual basis. Once we have the service in a completely verified, automatic mode of operation, it will provide a valuable addition to the information services we make available to our readers. □

Listing 1: Hard copy log of a session on the Robotics Information Service system. In this copy, all information entered by the user via the phone network has been underlined.

```
WELCOME TO THE ROBOTICS AGE ROBOT BULLETIN BOARD SYSTEM.

TYPE ANY LETTER: i


HOW MANY i CHARACTERS DO YOU SEE ON THE ABOVE LINE? (1 OR 2) 1
DO YOU NEED EXTRA LINE FEEDS WITH EACH CARRIAGE RETURN? n

WHAT IS YOUR FIRST NAME?  Reginald
WHAT IS YOUR LAST NAME?  Rogers
WELCOME TO THE WONDERFUL WORLD OF ROBOTS, Reginald

        ROBOT BULLETIN BOARD SYSTEM
            DO YOU WISH TO:

         R)EAD THE INFORMATIVE BULLETINS.
         S)UBSCRIBE TO ROBOTICS AGE MAGAZINE.
         C)HANGE YOUR MAILING ADDRESS.
         P)LACE A CLASSIFIED ADVERTISEMENT.
         Q)UIT AND LEAVE THE BULLETIN-BOARD SYSTEM.

CHOOSE: R,S,C,P,Q: S

NOW PREPARING TO ACCEPT SUBSCRIPTION INFORMATION.

COMPANY NAME:  Top of the Stack Innustries
ADDRESS 1:  2354-A East Aguila Highway
ADDRESS 2:
CITY:  North Nowher
STATE:  AZ
ZIP CODE: 85320
MASTERCARD OR VISA (M/V):  M
CARD NUMBER?  1365346389344
INTERBANK NUMBER: 1088
EXPIRATION MONTH: (1 - 12) 10
EXPIRATION YEAR: (82-84) 82
IS THIS NEW SUBSCRIPTION OR RENEWAL(N/R) n
HOW MANY YEARS: (1 - 3) 1


1)  LAST NAME: Rogers
2)  FIRST NAME: Reginald
3)  COMPANY NAME:  Top of the Stack Innustries
4)  ADDRESS 1: 2354-A East Aguila Highway
5)  ADDRESS 2:
6)  CITY: North Nowher
7)  STATE: AZ
8)  ZIP: 85320
9)  PAYMENT METHOD: MASTERCARD
10) CARD NUMBER: 1365346389344
11) INTERBANK NUMBER: 1088
12) EXPIRATION MONTH: 10
13) EXPIRATION YEAR: 82
14) NEW SUBSCRIPTION
15) NUMBER OF YEARS: 1

A)CCEPT OR C)HANGE INFORMATION: (A,C) c
ENTER NUMBER TO CHANGE: (1 - 15) 3

COMPANY NAME: Top of the Stack Industries

1)  LAST NAME: Rogers
2)  FIRST NAME: Reginald
3)  COMPANY NAME:  Top of the Stack Industries
4)  ADDRESS 1: 2354-A East Aguila Highway
5)  ADDRESS 2:
6)  CITY: North Nowher
7)  STATE: AZ
8)  ZIP: 85320
9)  PAYMENT METHOD: MASTERCARD
10) CARD NUMBER: 1365346389344
11) INTERBANK NUMBER: 1088
12) EXPIRATION MONTH: 10
13) EXPIRATION YEAR: 82
14) NEW SUBSCRIPTION
15) NUMBER OF YEARS: 1

A)CCEPT OR C)HANGE INFORMATION: (A,C) c
ENTER NUMBER TO CHANGE: (1 - 15) 6

CITY: North Nowhere

1)  LAST NAME: Rogers
2)  FIRST NAME: Reginald
3)  COMPANY NAME:  Top of the Stack Industries
4)  ADDRESS 1: 2354-A East Aguila Highway
5)  ADDRESS 2:
6)  CITY: North Nowhere
7)  STATE: AZ
8)  ZIP: 85320
9)  PAYMENT METHOD: MASTERCARD
10) CARD NUMBER: 1365346389344
11) INTERBANK NUMBER: 1088
12) EXPIRATION MONTH: 10
13) EXPIRATION YEAR: 82
14) NEW SUBSCRIPTION
15) NUMBER OF YEARS: 1

A)CCEPT OR C)HANGE INFORMATION: (A,C) a


NOW STORING INFORMATION.  PLEASE WAIT.

        ROBOT BULLETIN BOARD SYSTEM
            DO YOU WISH TO:

         R)EAD THE INFORMATIVE BULLETINS.
         S)UBSCRIBE TO ROBOTICS AGE MAGAZINE.
         C)HANGE YOUR MAILING ADDRESS.
         P)LACE A CLASSIFIED ADVERTISEMENT.
         Q)UIT AND LEAVE THE BULLETIN-BOARD SYSTEM.

CHOOSE: R,S,C,P,Q: Q

THANK YOU FOR CALLING ROBOTICS AGE MAGAZINE.
PLEASE HANG UP NOW.
```

Media Sensors

The Wall Street Journal April 6, 1982

In an article entitled "Household Robots May Come Next As Hobbyists Work on Prototypes," Richard A. Shaffer speculates on how long we may have to wait until the "personal" robot appears. Portia Isaacson, a Dallas consultant, believes the home-robot industry is in an important hobby phase, just as the home-computer industry was in 1975. The type of experimentation this entails will further develop robots in the home environment.

The long awaited Heath robot design system is rumored for release in 1982 or 1983. Although Heath will not reveal any specifics, speculation has it that it is a small, wheeled machine, with a mechanical arm, controlled by its own onboard computer. Sound waves will be used for obstacle detection. Estimated price is between $1500 and $3000. Several small companies are already selling robot arms with prices ranging from $695 to $2500. The availability of ready-made or easily assembled kits will help the home development of robots because many people who wish to design methods of controlling robots may not have the necessary skills to design and construct the physical hardware.

Constructing a robot in one's own basement can be costly. Glenn Hoffstatter spent $400 building Ambulatron, which just roams around the room. Charles Balmer, Jr. has invested $2000 in his robot, Avatar (an incarnation in human form according to *Merriam-Webster*), but estimates the cost could have risen to $100,000 if many of the parts had not been obtained as surplus. These robots do not perform the many wonderful functions science fiction has lead us to believe should be within the capability of robots: they move around, lift tool boxes, and some act as electronic sentries in the home. However, they do not make the bed, fix dinner, set the table, or take out the garbage.

The first personal robots will probably resemble the first personal computers — awkward and not very useful. Mrs. Isaacson predicts that once the machines are available, inventive experimenters will find many ways to control them and make them perform useful functions. Carl Helmers, editor of *Robotics Age* magazine, believes robot vacuum cleaners and lawn mowers will be among the first serious applications. Mrs. Isaacson thinks a robot security dog could retail for less than $500. "Take your cutest, most cuddly stuffed dog. Put wheels on its paws. Add sensors for noise and motion. Install an electronic chip that will bark, a computer that can recognize its owner's voice, and a motor to wag a tail. It would be great. You wouldn't have to feed or housebreak it."

Infoworld May 17, 1982 page 12.

This edition of *Infoworld* contained two very interesting news items. A report from International Resource Development (a market research firm) predicts more of the traditional electronics manufacturers will begin marketing robot systems. The report predicts small manipulative jobs such as electronic assembly and pharmaceutical packaging will provide a major robot market. Although the report is optimistic regarding the robot market, it advises caution due to the current recession.

Another *Infoworld* story describes a new company called Rodent Associates which plans to design and produce optical mice. The optical "mouse" is said to have a higher positional accuracy than the traditional mechanical mouse. An on board microcomputer frees the host computer of any necessary calculations which must be made to determine the mouse's position.

Electronic News April 12, 1982

Electronic News cites a Congressional study that says the United States is about to begin a robotics revolution and that government and public policy "should not be hostile toward development of robotics in the U.S." The Joint Congressional Economic Committee does not believe robots will cause considerable job displacement: rather, they believe that long-term robot development will create new job opportunities and better working conditions although a small number of people will be displaced in the short run. The study stated that "robots do have the potential of replacing blue-

Media Sensors

collar workers who belong to unions with white-collar workers (robot programmers for instance) who may not typically belong to unions. Thus, roboticization may come to be viewed by union leadership as union-busting."

A general consensus sees significant growth in the use of robots in the 1980s and beyond. One conservative estimate believes annual robot shipments will exceed 2500 per year by 1984. A more optimistic projection by the National Bureau of Standards and the Robot Institute of America sees shipments of 4800 per year by 1985 and 17,100 per year by 1990.

The greatest impact by robots may be in service industries. Developing "intelligent" robots that can perform a variety of tasks and operate in an office environment will be the key requirement.

BYTE Magazine
May 1982 page 286.

An article titled "Coln Robotics Armdroid: The Small-Systems Robot" provides a review of an inexpensive robot arm kit. The 5-axis arm is stepper-motor driven and can be controlled from a TRS-80 Model I (other computer interfaces are said to be in design). A feature which makes this small arm different from many others on the market is the 3-fingered "hand" which is standard.

Letters

The Rhino by a Nose

Dear Sir,

Mr. Harprit Sandhu's article on the Rhino XR-1 (*Robotics Age* March/April Vol. 4 No. 2) was outstanding.

His manual which at first he offered for $36 and more recently for $20 is something I had considered buying. At last I have an idea of what's in the manual! It will take me a few weeks (months) to figure what the Rhino arm can do — your *Robotics Age* article was/is very informative.

Ed Zorn
2830 W. Berridge Lane
Phoenix, AZ 85017

THE MICROBOT TEACHMOVER

John W. Hill, Ph.D., Vice President
and
Clement M. Smith, Research Engineer
Microbot, Inc.
453-H Ravendale Drive
Mountain View, California 94043

Until now, operating any of the low-cost tabletop robot arms on the market required the use of an external computer. If you didn't already own a computer, this increased your initial cost. And even if you did already own a computer, there was another drawback. Programs for robot arms had to be written in terms of motor steps, rather than actual arm motions. For example, to command the arm to reach for an object, the user couldn't simply specify where in space the arm should go, but had to figure out in advance which arm joints had to be rotated and by exactly how much. This constraint has proved awkward for many a user, thus defeating one of the major purposes of these low-cost arms: robotics education and investigation.

The TeachMover, designed and manufactured by Microbot Inc., is a tabletop robot arm which needs no host computer. It comes with a built-in microprocessor and a hand-held teach control which lets the user operate the arm and record arm motions without computer programming (photo 1). The teach control itself (photo 2) has virtually all the capabilities common to the control boxes used on large industrial robots costing upwards of $40,000.

The TeachMover allows the user inexpensive, hands-on experience with little initial training. The basic operations are simple. The unit comes with a complete user manual and with a demonstration program stored permanently in system firmware. The TeachMover is exceptionally safe to use, even by novices, since it operates at low power and since there is virtually nothing the user can do to damage the unit by using the controls incorrectly.

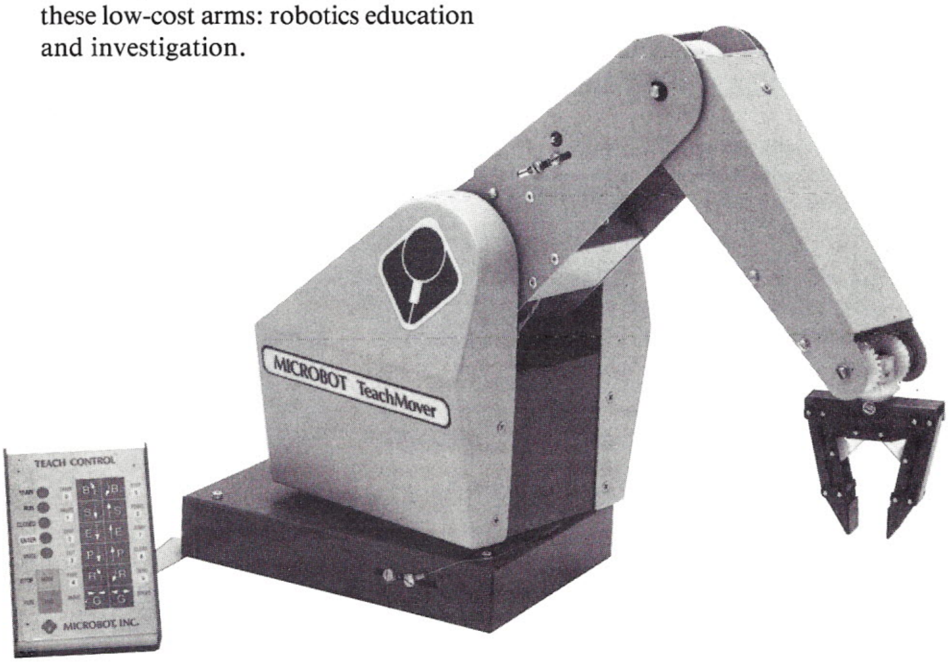


Photo 1: The TeachMover consists of a hand-held control box and a five-jointed, cable-driven mechanical robot arm. The arm, which has a microprocessor in its base, can also be controlled from an external computer using six simple commands.

Control by an external computer does, of course, have its place in industrial robotics. We have incorporated two RS-232C asynchronous serial communication lines so that the TeachMover arm can be controlled by a host computer or computer terminal as an alternative to the hand-held teach control. Even this mode of operation is unusually simple; only six commands are needed to accomplish all possible program options and arm motions when the TeachMover is in the serial interface mode. Additional commands allow programs to be downloaded from a host computer to the robot memory, or uploaded from the robot to a host. The latter capability means the user can develop arm motion programs on the teach control and then save them on disk.

On-Board Computer and Interface: The base of the TeachMover arm houses a 6502A microprocessor, which is an eight-bit, 2 MHz chip used to coordinate joint motions of the arm and handle all input and output. The 6502 is the microprocessor used in the Apple, Atari, and PET computers.

All internal system software programs are stored in 4K bytes of ROM. User-generated arm motion programs are stored in 1K of programmable memory; these programs can be up to

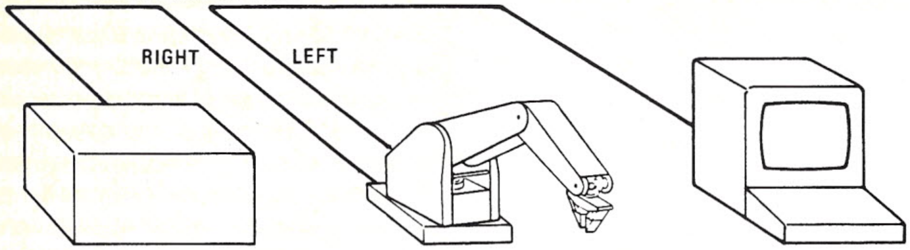


Figure 1: The TeachMover can be placed in series with a host computer and a terminal or printer. Signals from the terminal or printer to the host computer pass through the TeachMover unchanged. Signals passing in the other direction also pass unchanged, unless they are preceded by an @ sign. These latter signals are interpreted as TeachMover commands.

53 steps long. Memory can be expanded to extend the maximum program length to 126 steps.

The right-hand serial port in the base of the arm is used for interfacing with a host computer. The left-hand port is for a terminal or printer. Signals from the left-hand port always pass through to the right-hand port unchanged. The same is true for signals passing from the right port to the left port, unless these signals are preceded by an "at" sign (@), in which case, the signals are interpreted as arm commands.

The left-hand port is useful when the host computer itself has only one serial port. In such cases, the computer can still interface with the TeachMover without losing its ability to run a remote printer or terminal. This is accomplished simply by placing the TeachMover in series with the other peripheral (figure 1).

The rate of serial transmission may be set to 110, 150, 300, 600, 1200, 2400, 4800, or 9600 bps by means of switches located on the computer card inside the base of the TeachMover (photo 3).

Also included in the base of the unit is an auxiliary parallel I/O port which permits the user to interface the TeachMover to external equipment via a 16-conductor flat ribbon cable. Five user output bits may be set to logical 1 and cleared to logical 0 under program control to turn other equipment on or off when a given motion is complete. Seven user input bits may be read to initiate an arm sequence when a given external condition is met. These inputs can come from external switch closures or from TTL logic signals generated by a host computer.

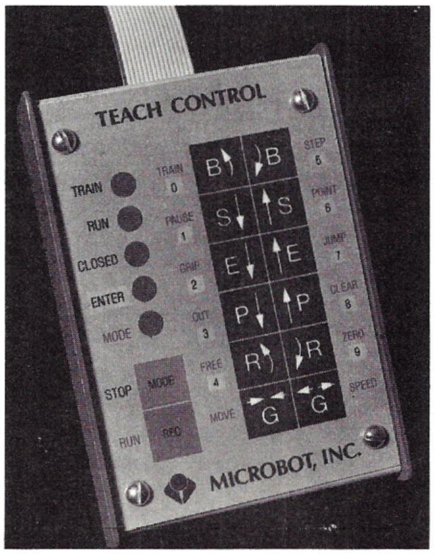


Photo 2: Using only 14 keys and five indicator lights, the teach control allows the user to independently move all arm joints; compose, step through, and edit arm motion programs; enter constants for speed of arm and duration of pauses; turn external switches on or off; and perform conditional branching based upon the state of the grip switch or specifiable external inputs.

Cable-Driven Design: The TeachMover arm (figure 2) is virtually identical to the arm on Microbot's MiniMover-5 robot (see "Introducing MiniMover-5," *Robotics Age*, Summer 1980, Vol. 2, No. 2). There are six drive motors: one each for base, shoulder, and elbow joint; one for hand opening; and one each for the left and right wrist gears. The wrist mechanism is a differential. When left and right wrist gears move in the same direction, the wrist motion is that of "pitch." When they move in opposite directions, the motion is "roll."

Many robot arms are designed with several of the drive motors mounted on the extension members (hand, forearm, and so on). This places an extra load on these members and increases their inertia. This, in turn, necessitates that the motors which move these members be stronger and more expensive than they would otherwise need to be.

In designing the arm for the MiniMover-5 and TeachMover, we decided to place all the drive motors in the body and to use a system of cables to manipulate the members. Our design represents an adaptation and refinement of the "tendon technology" used in aircraft, high-speed printers, plotters, and manipulators used in handling nuclear materials in "hot cells." To further reduce the number of mechanical parts in the arm, we placed all six drive gears on the same shaft. Putting all the motors in the body gave us an added bonus. The arm is very stable and does not have to be bolted down to keep it from tipping.

The drive gears are all operated by stepper motors which are driven under microprocessor control by digital power ICs (integrated circuits). We decided on stepper motors because of their low cost and the ease with which they can be controlled from a computer. We designed the drive system so that one motor step produces an incremental motion of .011 inch (0.25 mm) or less at the tip of the fingers. This incremental motion is hardly visible.

The cabling in the TeachMover is designed so that rotation of the shoulder or elbow joints does not change the orientation of the hand in space. Thus, if the hand is holding a container of fluid and the shoulder or elbow rotates, the fluid won't spill. Manually rotating the elbow does, however, affect the hand opening. Preventing this would require some complex and unwieldy cabling. We instead compensated for the effect by designing the firmware to automatically uncouple hand opening from elbow rotation. In the serial interface mode, the user needs to program the uncoupling, but this is accomplished through a simple formula.

A hand closure switch interfaced to the robot computer permits intelligent robot behavior. The cable that operates the mechanical hand passes over a

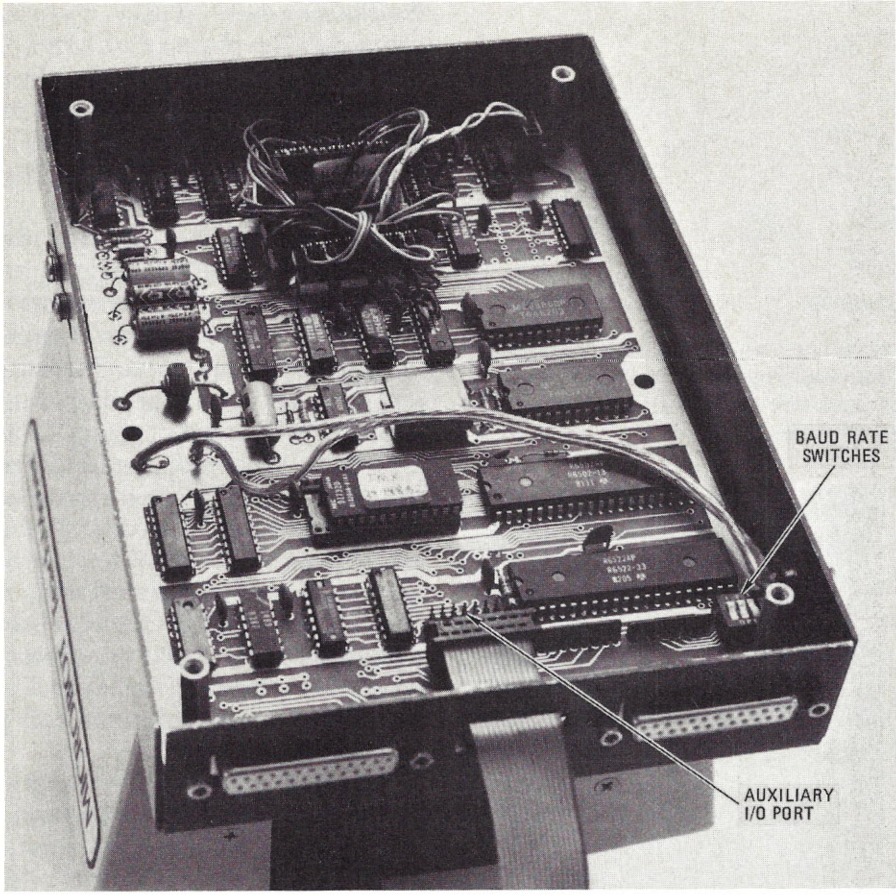


Photo 3: The computer card inside the base of the TeachMover contains the connector for the 16-conductor I/O parallel port cable and a set of switches which allow the serial transmission rate to be set to any of eight standard values from 110 to 9600 bps.

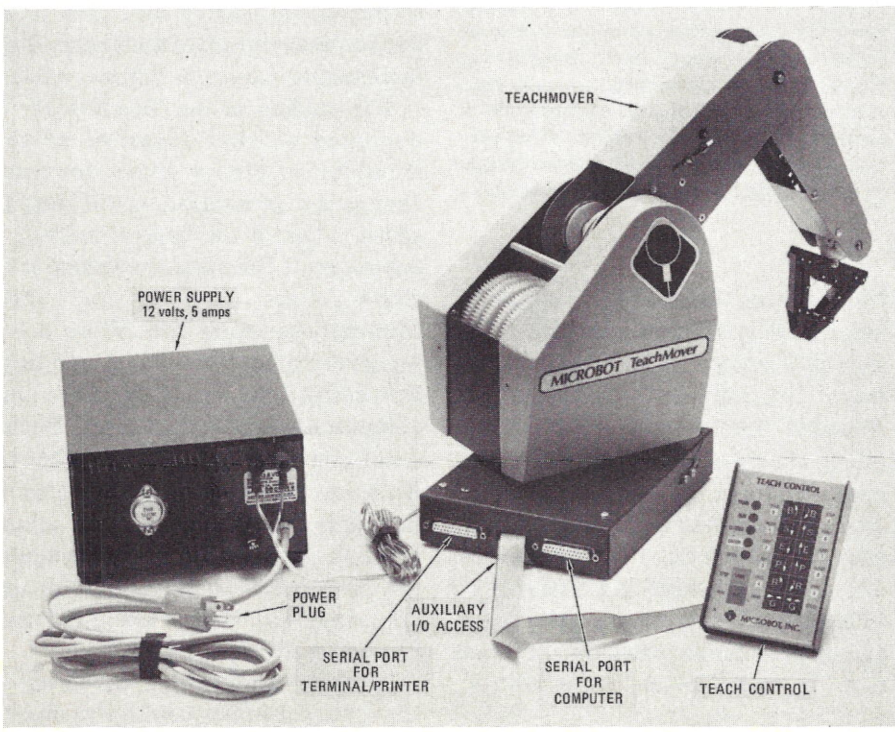


Photo 4: The base houses a 6502A microprocessor, two RS-232C serial communication interfaces, an auxiliary input/output parallel port, and connectors for the teach control and power supply.

switch located in the upper arm member (figure 3). This switch senses cable tension when the hand has closed either on an object or on itself. This switch allows the user to write programs to measure the size of objects, differentiate between objects of different sizes, and determine the presence of objects in the gripper. User control via the grip switch is supported by both the teach control (through a conditional branch command) and the RS-232C serial port.

Color-Coded Controls: Industrial teach controls vary in sophistication from devices which can only rotate the various joints to those with alphanumeric displays, diagnostic messages, and keypads that let the user specify arm motion via Cartesian coordinates. After weighing all the trade-offs between features and costs, we decided to include the following capabilities:

- independent movement of each joint by means of labeled control keys
- recording of movement positions to create a step-by-step, repeatable program
- ability to move the arm through a program one step at a time
- program editing
- specification of speed of arm and duration of pauses
- conditional branching based on the state of the grip switch or other external inputs
- ability to turn external outputs on or off (for example, to control other machinery)
- ability to manually position arm when power is off

In addition, we wanted to implement all these functions with a minimum number of control keys, again to keep product cost as low as possible. We decided to use three colored "overlays" so that the same keys could be used (1) to select functions, (2) to input numeric values, and (3) to position the arm. Rather than use an expensive alphanumeric display to indicate which overlay is in use, we developed a simple system of color-coded indicator lights and key labels.

When the red MODE light is on, the words printed in red apply to the keys,

and the user can select from TRAIN, STEP, PAUSE, RUN, and other functions. When the yellow ENTER light is on, the yellow numerals next to the keys apply, and the user can enter numerical values. The labels printed on the keys themselves apply when the teach control is in the TRAIN mode (or in the MOVE mode, as explained later).

Control Functions: The TeachMover incorporates 13 different control functions. They work as follows:

TRAIN and *RUN*: When the TeachMover is first powered up, the teach control is in the TRAIN mode, and the green TRAIN light is on. In the TRAIN mode, each of the key pairs B, S, E, P, R, and G control one of the joint motions of the arm (B = base, S = shoulder, E = elbow, P = pitch of wrist, R = roll of wrist, and G = grip). The keys in the right-hand column cause these joints to move in the direction indicated on the key face, while those in the left-hand column cause them to move in the opposite direction.

When the user presses one or more of these keys to move the arm to a desired position, he can record this position as a program step by then pressing the REC key. Up to 53 steps may be programmed; these steps are internally numbered 0-52. Pressing the REC key overwrites the current program step and then increments an internal sequence pointer so that the TeachMover memory is ready to record the next step. To run a program, the user first presses the MODE key (to exit from TRAIN) and then presses RUN.

PAUSE: The user can program pauses into a program by pressing the MODE key, if the MODE light is off, and then the PAUSE key. Once PAUSE is pressed, the yellow ENTER light comes on, and the two rows of keys may be used to enter numerals according to the yellow labels. The user simply enters a number (0 to 255) equal to the length of the desired pause in seconds, then presses the MODE key again. The PAUSE command is saved as a program step, and the sequence pointer is incremented. When the program runs, the arm pauses for the desired number of seconds.

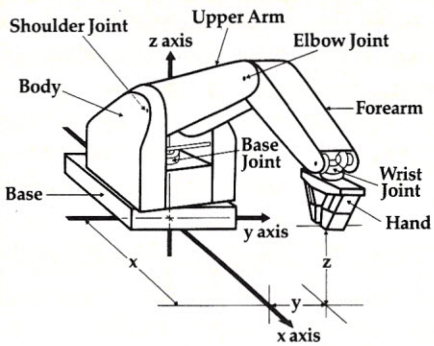


Figure 2: The TeachMover arm consists of a base, body, upper arm, forearm, and hand, separated by joints labeled as shown. To reduce the load at the extremities, all motors are mounted in the body.

SPEED: This command allows the user to change the speed of the arm. The SPEED command is not saved as a program step nor does it cause the sequence pointer to change. Issuing this command simply causes all subsequent recorded steps and manual motions to be executed at the commanded speed.

After the SPEED key is pressed, the yellow ENTER light comes on, and the user may enter any number from 0 to 15. Zero is the slowest speed, and 15 is the fastest. The TeachMover is always initialized to speed 6 when it is turned on. The correspondence between speed numbers and steps per second of the drive motors is given in a table in the user manual.

STEP: This command is useful during program development, for it allows the user to move the arm through a program one step at a time.

JUMP: This command allows the user to write highly sophisticated programs, for it provides for conditional branching. When the JUMP key is pressed, the yellow ENTER light comes on and the user enters two numeric values, pressing the MODE key in between. The first value represents the jump condition, and the second is the sequence pointer number (step number) to jump to if the jump condition has been met. The jump conditions are as follows:

- condition 0: grip switch is open
- condition 1-7: user input bit 1-7 is on (that is, set to 1)
- condition 8: never
- condition 9: always

For example, to cause the arm to go to step 5 in a program on the condition that the grip switch is open, one would key in: JUMP 0,5. Input bits 1-7 can be set in a variety of ways — for example, by sensor switches the user might mount on the TeachMover's fingers, by switches on external machinery, or by signals generated in a host computer program.

POINT: This frequently used command is similar to an unconditional jump. POINT 6, for example, means to go to step 6 in the program and proceed from there. Unlike the JUMP com-

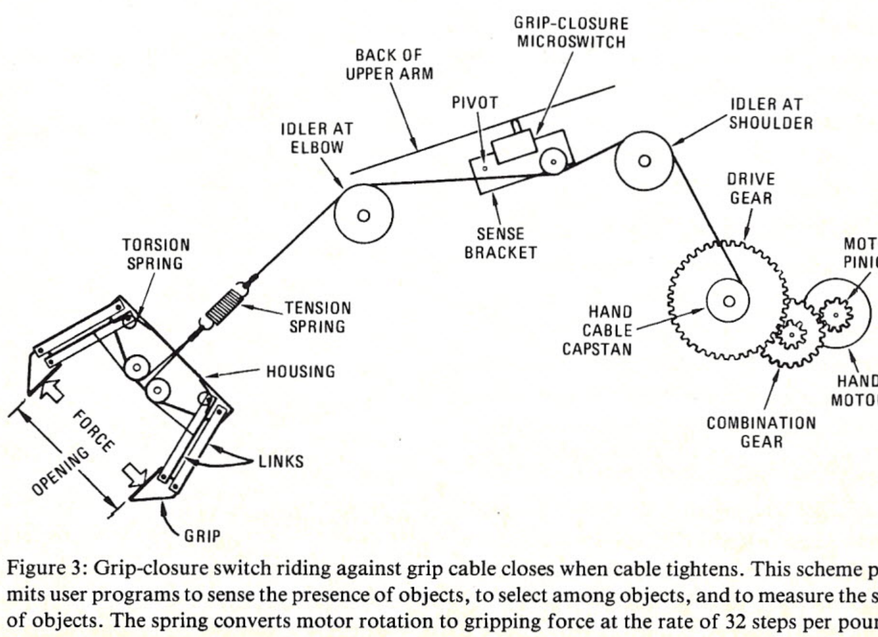


Figure 3: Grip-closure switch riding against grip cable closes when cable tightens. This scheme permits user programs to sense the presence of objects, to select among objects, and to measure the size of objects. The spring converts motor rotation to gripping force at the rate of 32 steps per pound.

mand, however, POINT does not create a program step. POINT is used simply to move to a given step in an existing program; it can be invoked even in the middle of program execution by simply pressing the MODE key first.

The POINT command is especially useful during program editing. To change a program step, the user need only POINT to the step in question, press the TRAIN key, use the joint control keys to achieve the desired new position, and then press REC. The user can also STEP through the program until the arm attains the position to be modified, then use TRAIN as before, except that stepping to a JUMP executes the JUMP.

The POINT command can also be used to execute multiple programs stored in memory. Program 2 in figure 4 for example, can occupy memory steps 21 through 31, with step 31 being an unconditional JUMP back to step 21. Program 2 can be executed simply by issuing the command POINT 21 prior to pressing the RUN key. Program 3 can be executed in a similar fashion. The POINT command is useful to run the demonstration program that is permanently stored in TeachMover firmware at location 126.

The POINT command has still another function. In creating a program, it is sometimes desirable to copy a program step; that is, key in a program step which causes the arm to move back to a previously achieved position. To do this, the user need only STEP through the program until the desired position is achieved. POINT to the program step to which that position is to be copied, then press TRAIN and REC. This duplicates the desired position at the desired program step.

CLEAR: This command is activated by first pressing the MODE key and then, while holding down the MODE key, pressing the CLEAR key. This clears all recorded positions and operations in program memory and sets the sequence pointer to step 0.

GRIP: This command causes the gripper to close by proceeding 32 motor steps past the point at which the grip closure switch is activated. This builds up about 1 pound of gripping force.

MOVE: This activates the joint control keys used in TRAIN mode but does not change the internal position registers or allow a position to be recorded. The MOVE command is useful in moving the arm back to a known position in the event of motor slippage or mechanical interference from an external obstacle.

FREE: This turns off all motor currents, allowing the arm to be positioned manually. Positioning the arm by hand is often faster than using the keys.

ZERO: This command is activated by first pressing the MODE key and then, while holding down the MODE key, pressing the ZERO key. This sets the TeachMover's six internal position registers (one for each motor) to zero and sets the sequence pointer to zero. The ZERO command can be used for program initialization.

OUT: This command allows the user to output a binary logic 0 or 1 on up to five output lines or to turn lights on the teach control box on and off, based on arm positions achieved or conditions met. The OUT command must be followed by two numerical entries; the first is an output number, and the second is a 0 or 1 (off or on, respectively, in the case of the teach control lights). The output numbers are as follows:

- output 0: the MODE light
- output 1-5: user outputs on the I/O connector
- output 6: the TRAIN light
- output 7: the RUN light
- output 8: the ENTER light

TeachMover

Description	Includes teach control, power supply and RRA-2 user manual
Configuration	Five revolute axes and integral hand
Drive	Electrical stepper motors — open loop control
Controller	Microprocessor with 4K bytes of EPROM and 1K bytes of RAM located in base of unit
Interface	Dual RS-232C asynchronous serial communications interfaces (data rate is switch selectable between 110, 150, 300, 600, 1200, 2400, 4800, and 9600 bps)
Program Languages	ARMBASIC through serial port
Teach Control	13-function keyboard
External I/O	5 output and 7 input bits under computer control
Power Requirements	12 to 14 volts, 4.5 amps DC (cont.)
Cable	Teach control cable length 3 ft. (900 mm)
Payload	1 lb. (454 gm) max. at full extension
Resolution	0.011 in. (0.25 mm) max. on ea. axis
Gripping Force	3 lbs. (13N) max.
Reach	17.5 in. (444 mm)
Static Load Force	4 lbs. (18 N) max.
Arm Weight	8 lbs. (4 kg)
Positioning Accuracy	±.030 in.
Speed	(0-6.5 ips) 16 selectable movement rates via teach control
Gripper Opening	3 in. max. (0-75 mm)
Control Method	PTP (Point to Point)
Programming Capacity	53 stored positions
Base Motion	±90°
Shoulder Motion	+144°, −35°
Elbow Motion	+0°, −149°
Wrist Roll	±180°
Wrist Pitch	±90°
Gripper	Included
Power Supply	Included. 105-125 VAC input, 13.8 VDC output, 5 amps. 220 VAC also available.
Application Manual	Included with purchase, also sold separately
Plastic Carrying Case	Optional

Table 1: TeachMover performance characteristics.

Sample Programs: A "pick and place" program which uses just the joint-control keys is shown in listing 1. Once the user masters the basic arm motions through creating simple programs like this, he can progress to more complex tasks. One such task is flowcharted in figure 5. Here, two different blocks are placed in predetermined locations. (P1 and P2 as shown in figure 6.) The arm does not know in advance which block is larger. The task is for the arm to locate the larger block, move it to a new location (P3), and stack the smaller block on top of it (location P4). Listing 2 shows how this is accomplished. Note how the JUMP command is used for conditional branching.

Serial Interface Commands: When the TeachMover is linked to a host computer over serial transmission lines, six commands can be issued to allow complete control of arm functions. These commands are:

@STEP: This causes all six stepper motors to move simultaneously. The syntax of the @STEP command is:

@STEP (SP), (J1), (J2), (J3), (J4), (J5), (J6), (OUT)

(SP) is a value which gives the speed of motion, (J1) through (J6) are the number of steps that each of the six motors are to be moved (J1 = base joint, J2 = shoulder, J3 = elbow, J4 = right wrist, J5 = left wrist, J6 = hand), and (OUT) is a decimal number which translates into the binary pattern the user wishes to appear at the user outputs mentioned above. (Note: the speed values, SP, are not the same as those used with the teach control, but are related to the latter by a simple formula.) For example, to rotate the base joint 20 steps at a speed value of 50 without rotating any of the other joints or changing any of the user outputs, the command would be:

@STEP 50, 20, 0, 0, 0, 0, 0

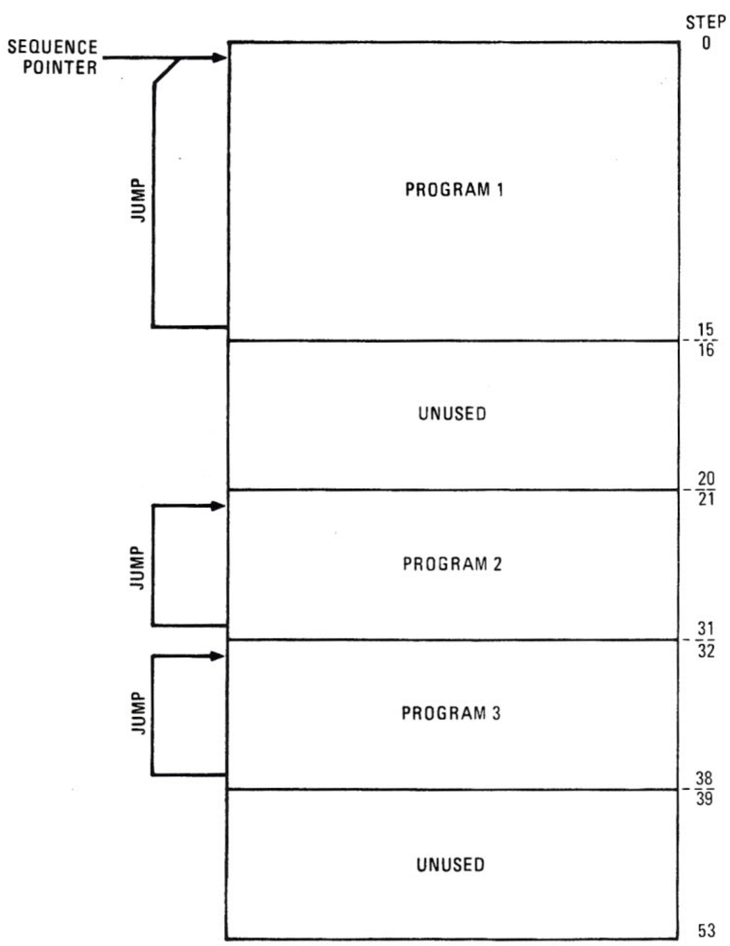


Figure 4: JUMP and POINT instructions can be used to store and execute several isolated programs within the TeachMover's 53-step memory.

or simply:

@STEP 50, 20

When the @STEP command is given, the TeachMover returns a 1 or a 0 to the host computer, depending on whether the command has been executed or not. The host computer can easily be programmed to test for this and to print INVALID COMMAND in the event of a 0.

Programming the TeachMover on a host computer does require knowledge of how rotation of the various motors affects motion of the arm itself. In addition, pitch and roll cannot be programmed directly, but only by specifying the proper coordinated motion of the left and right wrists, (J4) and (J5). A command to move the elbow joint affects the hand opening unless the user compensates by adding the elbow position to the hand position when the command is issued. Thus, if the desired number of steps for the base, shoulder, elbow, pitch, roll, and gripper are represented by B, S, E, P, R, and G, respectively, the correct motion command would be:

@STEP (SP), B, S, E, (P+R)/2, (P−R)/2, E+G

In designing the TeachMover, we programmed the firmware so that motion of the six motors is simultaneous rather than sequential. If unequal numbers of steps are specified for different motors, the firmware automatically coordinates the timing to produce smooth motion.

@CLOSE: This command causes the hand to close just until the grip switch is activated. The syntax is simply:

@CLOSE (SP),

The optional (SP) determines the speed of closing. The arm responds with 0 if there is a syntax error or with 1 if the gripper has closed.

@SET: This puts the arm into the TRAIN mode, activating the keys on the hand-held teach control. The syntax is:

@SET (SP)

This mode is terminated when the REC or MODE key is pressed on the teach control. If the REC key is pressed after

an @SET command, no step is recorded. The @SET command gives the user the option of programming arm positions via the teach control even in the course of execution of a host-driven program. After terminating this command, the arm responds with a 1, or a 0 if there was a syntax error.

Three additional commands allow entire programs to be transferred between the host computer and the TeachMover:

- *@QWRITE (SS)*: Downloads a sequence step to the TeachMover.
- *@RUN (SS)*: Starts running a recorded sequence at step (SS).
- *@QDUMP*: Uploads a program to the host, after first converting the program to ASCII format. Once uploaded, the program can, of course, be stored on disk. Thus, the user can develop programs on the hand-held teach control and save them for later use and investigation.

Room to Experiment: The TeachMover allows the user to try simple experiments at first, then progress to more complex investigations as he gains experience. One interesting experiment involves connecting one of the user input bits to one of the output bits. This provides the user, in effect, with a one-bit memory: A condition can be read at one time and acted upon at a later time.

Sensors may be installed in the fingers of the robot and connected to the input bits to obtain more intelligent operation. Microswitches or optical (phototransistor-LED) sensing modules will permit the robot to sense objects before grasping them. An optical retro reflector will permit the robot to scan bar codes or to align itself using a

STEP	DESCRIPTION
0	Home position (gripper open)
1	Move right
2	Move down
3	Close gripper
4	Move up
5	Move left
6	Move down
7	Open gripper
8	Move up (to clear object)
9-52	(Null)

Listing 1: A simple "pick and place" program. With the teach control in the TRAIN mode, the user moves the arm from one position to the next, using one or more of the arm motion keys. The REC key is pressed after the arm achieves each desired position. When the program RUNs, the arm cycles through all the steps repeatedly until another command is given.

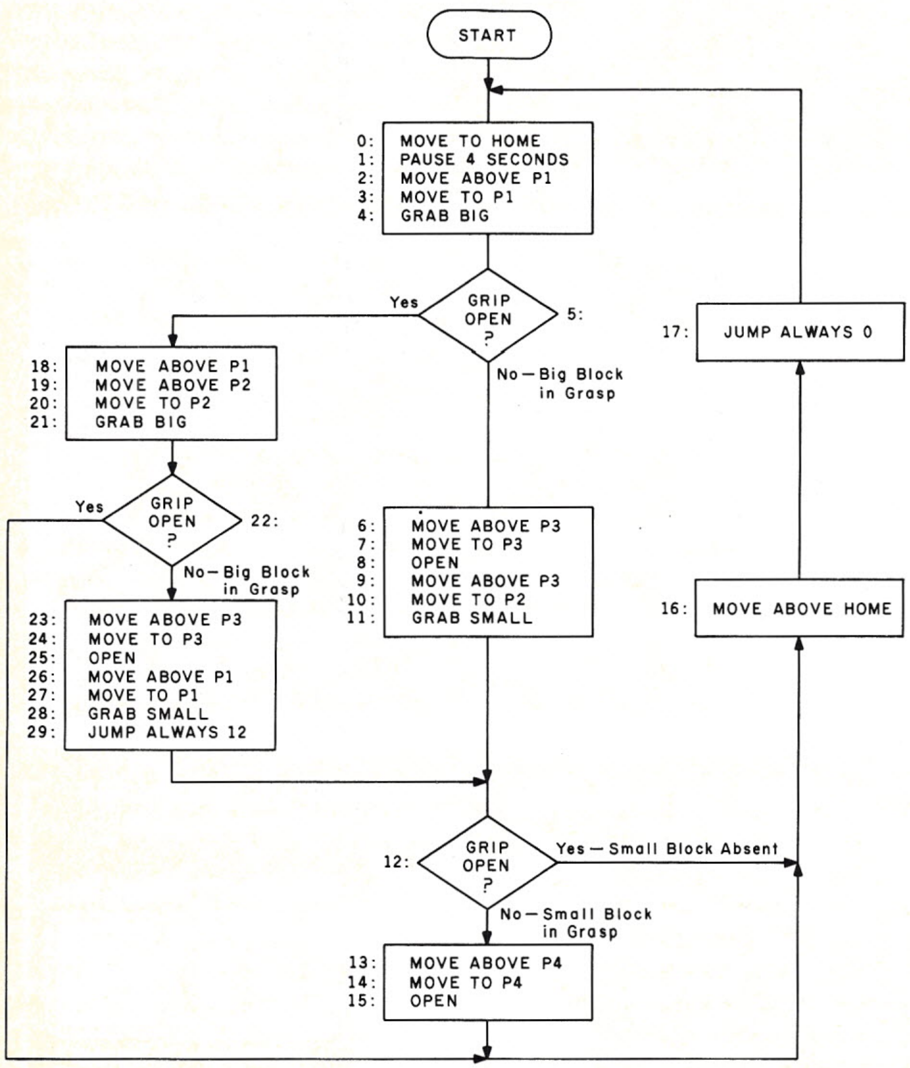


Figure 5: Flowchart for a block-stacking program. The arm reaches for one of two different blocks and determines whether it is the smaller or the larger one. If it is the larger block, the arm moves the block to a new location and then stacks the smaller block on top.

STEP	DESCRIPTION
0	Move to home
1	Pause four seconds
2	Move above P1
3	Move to P1
4	Grab big
5	Jump if grip open to 18
6	Move above P3
7	Move to P3
8	Open
9	Move above P3
10	Move to P2
11	Grab small
12	Jump if grip open to 16
13	Move above P4
14	Move to P4
15	Open
16	Move above home
17	Jump always 0
18	Move above P1
19	Move above P2
20	Move to P2
21	Grab big
22	Jump if grip open to 16
23	Move above P3
24	Move to P3
25	Open
26	Move above P1
27	Move to P1
28	Grab small
29	Jump always 12

Listing 2: This program accomplishes the task described in figures 5 and 6. The necessary conditional branching is accomplished by means of the JUMP instruction.

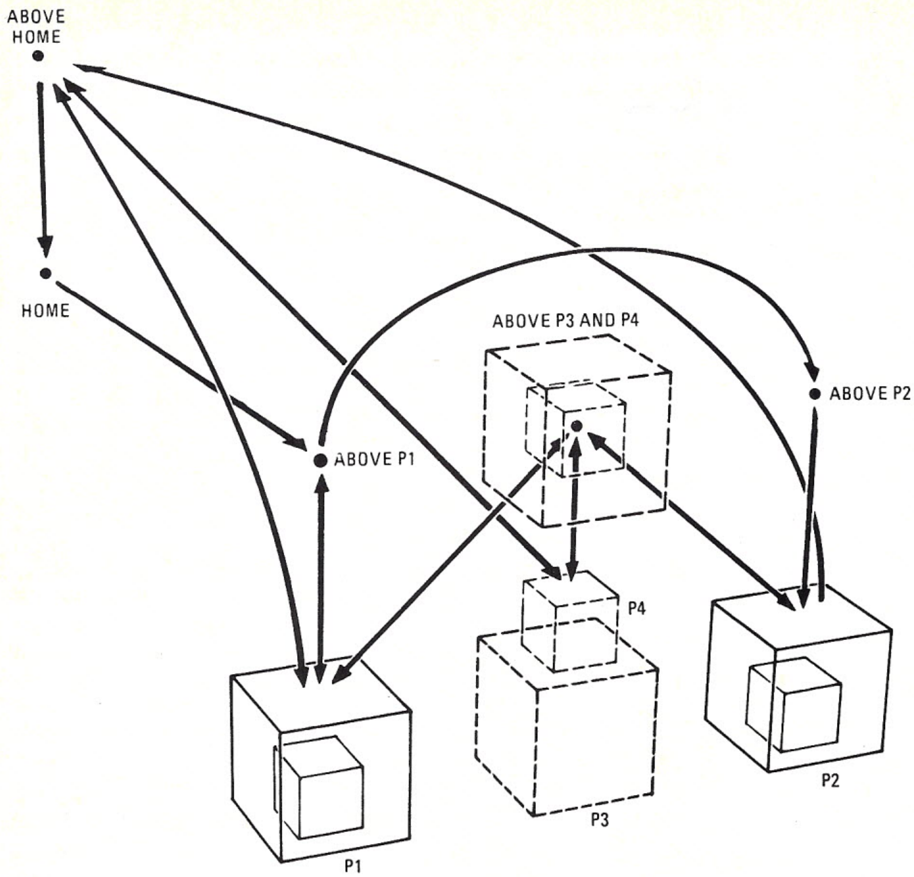


Figure 6: Definition of position referred to in block-stacking program with motion shown by arrows.

reflection target. A light beam from finger to finger also offers interesting possibilities.

Connecting two TeachMover arms together with a 16-conductor flat cable offers another possibility for investigation. Conditional branches can be used, for example, to cause one arm to take an object from the other arm.

Other Products: The TeachMover is available now from Microbot, Inc., 453-H Ravendale Drive, Mountain View, California 94043; telephone 415/968-8888.

Microbot is in the exclusive business of developing and manufacturing robots. In addition to robots for education and experimentation, Microbot is initiating manufacturing of small industrial robots that are compatible with the TeachMover. Soon to be released in a test marketing program is the Microbot Alpha, made entirely of industrial-grade parts. As compared with the standard TeachMover, the Alpha model will have both greater speed and greater lifting power. □

SOME NOTES ON THE RHINO XR-1 AND MINIMOVER 5

Jorg R. Jemelka
Digital Computer Laboratory, Room 29G
1304 West Springfield Avenue
Urbana, Illinois 61820

During the summer of 1981, the microcomputer laboratory at the Urbana-Champaign campus of the University of Illinois acquired a Minimover-5 instructional robot. The laboratory, headed by Mike Faiman, and the artificial intelligence group, under Donald Michie, undertook a joint project integrating elements of an ongoing vision project, a suitable task domain, a problem solver, and the Minimover-5. At the same time, the first Rhino XR-1s went into production locally. Two early models were acquired for inclusion in the project, one on loan, the other purchased.

The Minimover-5 and the Rhino XR-1 are articulated (anthropomorphic) manipulators. In both, a change in the angular displacement of a joint does not influence the orientation of the other joints in workspace coordinates, but does affect their relative orientations. Joint transformations are thus performed in Cartesian space, using two triangles as a geometrical model (see figure 1). The first is the right triangle formed by the shoulder-wrist extension-wrist height (ABC), the other is the shoulder-elbow-wrist triangle (ADC). The angular displacements are translated into motor steps and passed to an assembly-language driver. The driver attends to the details of issuing individual motor commands.

The project world consists of a table setting with plate, knife, fork, spoon, and cup. It is scanned, its components are identified, their relative positions and orientations are ascertained, and a determination of *correctness* is made. A *fix* is generated for an incorrect scene, which is then executed by the manipulators. The world model in which the manipulators operate is essentially two-dimensional. All of the objects have a Z coordinate of 0, except the cup, which has a significant Z. To operate in this world, approach, retreat, and rotation take place at shoulder height; grasping and releasing occur at object height. This pick-and-place task, analogous to those performed by larger machines in industry, is a suitable test of the manipulators' capabilities. A typical move sequence would be:

1 Raise hand to shoulder height,
2 Extend hand over object XY,
3 Open hand and align with object,
4 Descend to object Z and close hand,
5 Raise object to shoulder height,
6 Extend hand to destination XY,
7 Rotate to destination orientation,
8 Lower to object Z,
9 Open hand,
10 Raise hand to shoulder height at destination XY, close hand,
11 Park at "home" position.

The apparent similarity of the two machines, when seen from the task definition level of abstraction, masks significant differences that influence implementation strategies.

Rhino XR-1. The Rhino XR-1 is powered by servomotors with feedback from incremental encoders that count steps. All communication from the host computer to the Rhino's Intel 8748 control microprocessor is serial. Motor control is performed using the Bang-Bang approach, full voltage is applied to a motor until its step count register is decremented to zero. There are six encoder steps per motor revolution. Dynamic braking is provided, locking the motor at the encoder position attained when its register is fully decremented. The 8748 reserves a memory location for each motor counter. Since the sign bit acts as a direction flag, individual commands are limited to 127 steps forward or reverse. Fifteen commands are required to move a typical joint 180 degrees.

Although the presence of a dedicated control processor holds the promise of reducing host processing time, the considerable range of motion required of each joint by our project application necessitated the incorporation of an assembly-language intermediate driver. This program receives as parameters the motor code (ASCII characters A through F) and the number of steps to be driven. The program issues repeated commands of 100 steps, first reading the feedback registers to ensure that it contains 20 or fewer steps, lest the addition of the new step sequence affect

the direction bit and reverse the motor. Remaining steps fewer than 100 are transmitted as a final command, after which the intermediate driver returns control to the coordinate transform program.

While developing the driver routines, we discovered that turning on more than three motors simultaneously frequently resulted in a lowering of the power supply voltage to under the amount required to sustain the on-board logic. This voltage drop would cause uncontrolled acceleration of all joints, requiring a system shut-down to prevent damage to the motors. A separate 5 V supply was added to power the logic. The current production series of Rhino XR-1s is provided with an extra 12 V power supply for the 8748 and encoder LEDs, which should eliminate the problem.

Moving the various Rhino XR-1 joints through their full range of motion revealed a number of errors in the motor steps to angle conversions as stated in the manual. Tests were run using a construction level to align the elbow and hand horizontally and vertically. Using this procedure, ±2 steps deviation from the horizontal or vertical could be detected. The constants derived are:

Steps/Radian

	Derived	**In Manual**
Hand	716.19785	750.57471
Elbow	725.74715	750.57471
Shoulder	500.38356	500.38356
Base	420.1694	359.69017

The rigidity of the chain driven shoulder joint accounts for the direct translation of motor steps through the various reductions to shoulder angle.

Our early series Rhino XR-1 came equipped with the "standard" hand. The finger assembly is operated by two concentric shafts. The outer shaft, on which the assembly is mounted, is used for rotation. The threaded, inner shaft opens and closes the fingers. Rotational accuracy for the standard hand is 15 degrees per encoder slot, or 3.81971 steps/radian. Closing or opening the fingers full throw requires approximately 7 seconds. Testing the finger assembly revealed that repeated open/close cycles dislodged the threaded actuating shaft from its mounting, a situation since remedied in later models. The open/close shaft is now dimpled to provide a secure seat for its set-screw. A further complication results from the fact that both the axis of rotation and the open/close shaft are concentric. The hand tends to open or close further on rotation, since it rotates around the threaded, open/close shaft. The optional "deluxe" hand is designed to decouple these axes.

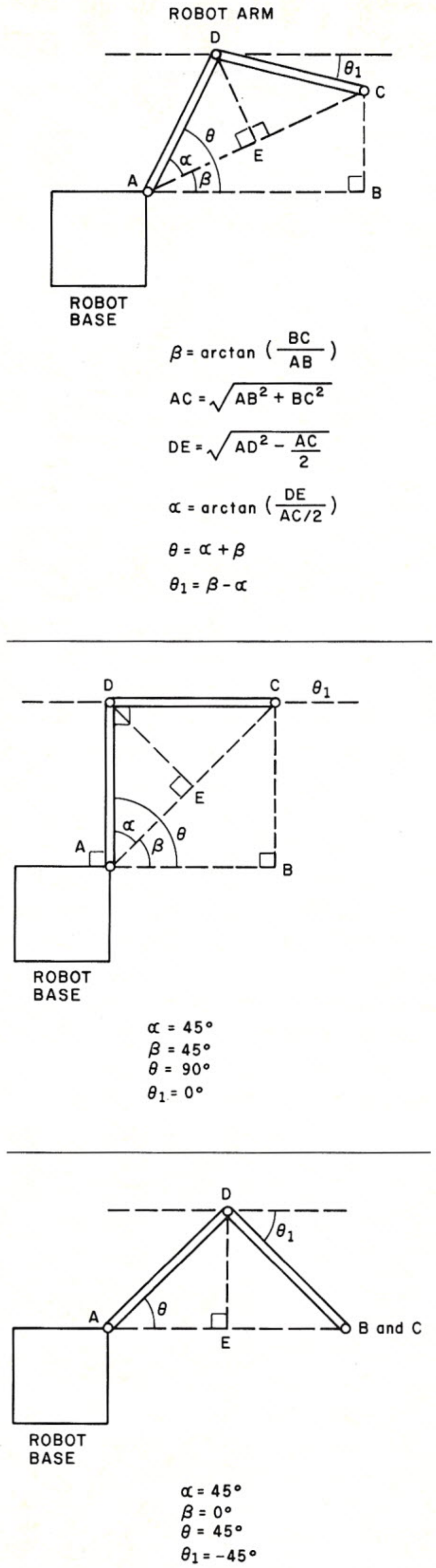


Figure 1: Some of the relationships between angles and joint placement in an articulated arm. All joint and manipulator placement calculations can be made using the simple trigonometric relationships defined in figure 1a. The lengths of arm segments AD and DC will remain constant for any specific arm. Figures 1b and 1c show values for two specific cases.

Minimover-5. The Minimover-5 is powered by stepping motors without feedback. Control is maintained by the host computer through an assembly-language driver, full documentation for which is provided in the Minimover-5 manual. Communication is parallel, each byte containing a latching bit, a motor address, and a phase pattern. Eight successive phase patterns, each a 4-bit binary number, rotate the motor shaft 3.75 degrees. The problems encountered interfacing the Minimover-5 to our Northstar Horizon were minor. The early manual had misprints which incorrectly labeled the connector pins (since remedied). A timing problem with the latching pulse was cured by sending address and data out with the latch bit set to logical 1, sending it out again with the latch bit 0, and then setting the latch pulse and sending the address and data a third time. The only change from the interfacing procedure as outlined in the manual is that the initial transmission is sent with the latch bit high.

The Minimover-5 driving software accepts 16-bit integers as commands. This allows straightforward command sequences to move all joints their full range. The Minimover-5 driver algorithm also simulates speed control, ensuring that all motors start and stop simultaneously. Implementation consists mainly of translating the Zilog mnemonics in the manual to Intel format.

High-level Programming. The high-level coordinate transform program to calculate the step sequence required in moving the various objects was written in BASIC. The similarity of the two

machines simplified the programming task considerably. Aside from dimensional and motor constants, only the hand geometrics required different implementation strategies. The programming task was further simplified by the assumption, allowable in this case, that the hands always point straight down.

The Minimover-5 is designed so that its *home* position (shoulder fully back, elbow at rest on shoulder, hand normal to table surface), leaves its fingertips 35 mm ahead of and 10 mm below the shoulder joint. This provides values suitable for the calculation of initial and subsequent shoulder and elbow angles as well as providing fixed points for the end-of-travel of all joints. The Rhino XR-1's similar home position leaves the fingertips slightly above and imperceptibly ahead of the shoulder joint. Attempts to base calculations on these barely measurable initial values resulted in final position errors that varied with the joint extension. Consequently the defined home position was moved so that the fingertips were centered below and significantly ahead of the shoulder joint. This served to eliminate joint calculation error, with the drawback of leaving the home position in free space. The nearest limit switch cam mounting points are intermediate to this position and are not used as home position indicators, although they could, with additional software, be used to guide the arm to its parked position.

During the development of the high-level software, minor errors in the Z-axis were compensated for by changing the fingertip to table height constants of both machines from the measured values. In this final development phase, collision with the table surface occurred occasionally, resulting in a loss of positional information, since motor steps, rather than joint angles, are controlled. The Minimover-5 lost its position more easily because of its weaker stepping motor drive, while the Rhino XR-1, because of its rigid shoulder chain, suffered a positional error mostly in the elbow joint.

Reaching objects close to the base of the Rhino XR-1 often resulted in unwanted collisions, since the shoulder and elbow joints are not coordinated to reach their destinations at the same time. In cases where the shoulder joints have few steps and the elbow many, the shoulder, reaching its goal first, provides a pivot for the elbow, low enough to cause the elbow to collide with the table while it is in the process of finishing its move. This could be compensated for by segmenting the moves into smaller subgoals to be attained en route to the final destination. The Rhino XR-1 has sufficient reach, allowing us to keep its work space 100 mm out from the base without suffering a real penalty.

Testing. Both machines were given a move sequence to execute repeatedly under program control. The first test of twenty-five runs yielded timing and repeatability information. The overall cycle time to move both machines from home to 200,200 mm XY and 200,-200 mm XY was 1 minute 43.14 seconds. The Rhino XR-1 required 1 minute 05.18 seconds, the Minimover-5 34.25 seconds, keeping in mind the 7-second hand open/close time. At cycle 23, the only error encountered in either machine was a 45 degree wrist rotational error, a result of round-off errors. A second test of 50 cycles resulted in the Rhino XR-1's slipping its shoulder drive belt on cycle 33 while parking. This overshoot error was in the direction of the shoulder joint, caused by the loss of a 100-step command due to the absence of a serial communications protocol. Because of the shoulder's rigidity, this did not affect its performance on subsequent cycles. No errors were encountered with the Minimover-5 in either test.

With the completion of these tests, the hardware and software is fully operational. Project members anticipate further refinements with the development of C and Pascal software, the implementation of speed control, and a flexible driving routine for the Rhino XR-1's 8748 microcomputer. □

REFERENCES

Hill, J.W. "Introducing Minimover-5," *Robotics Age*, Summer 1980, pages 18 to 27.

Sandhu, H. "The Rhino XR-1: A Hands-on Introduction to Robotics," *Robotics Age*, March/April 1982, pages 10 to 18.

PATENT PROBE

Russ Adams
3008 Mosby Street
Alexandria, Virginia 22305

A patent is a contract between the Public and an Inventor. In exchange for the seventeen year right to exclude others from making the invention, the inventor, in addition to meeting other legal requirements, must fully disclose his invention to the Public. A patent, therefore, is not entirely a technical publication. It is, in part, a legal document. And legal documents tend to be written in that strange, precisely defined sub-language which the legal profession finds so appealing. Unfortunately, the technological gems contained in U.S. patents are often hidden from the scientist's view amid the legalese.

In addition to the legal semantics problem, the scientist may find a sketchiness of disclosure in patents. While the inventor is obliged to fully disclose his invention, he need only disclose it in terms that would enable one knowledgeable in the technology as defined by all patents and publications in the technology to make his invention. The scientist new to an area of technology may well start reading all the technical journals he can. But, he may still find himself lost when reading a patent because he has not read all the patent literature in the technology. Patents are the only source of technical information in many areas. Video arcade games are an example.

Yet patents are not the most accessible technology information resource. The primary repository is the U.S. Patent and Trademark Office located in Arlington, Virginia. More than 32 million paper documents are stored throughout two eleven-story office buildings, and organized in an elaborate classification system. Needless to say, finding information about a specific area of technology requires expert search assistance.

More than one thousand patents are issued each week. Very few could be classed as "breakthroughs." Most represent the unglamorous steady progress of man's ingenuity. We will examine these bread-and-butter inventions more closely.

The Orientation Problem. Sometimes the best solution is the simple solution. We find evidence of this in the patent to Jerry and Kerry Kirsch (U.S. Patent 4,187,051) issued on February 5, 1980. The invention provides a unique mechanical solution to a problem encountered in providing an industrial robot with sight. When video cameras have been used to allow robots to see, the fixed positions of the cameras have required complex trigonometric processing to rotate the video image of an object to a recognizable orientation. The processed signal is compared to learned images of the object and action is taken by the robot based on the result of this comparison.

The Kirsches' invention eliminates the complex trigonometric video signal processing by the simple expedient of rotating the video camera under computer control until the orientation matches the learned image.

The device consists of a servo motor driven xy carriage. Mounted on the carriage is a piston with a rotating and reciprocating horizontal arm. One of two video cameras is mounted on the arm. The camera is a narrow angle high-resolution-type like Model TN2200 manufactured by the General Electric Company. An inclined reflector is mounted at the end of the moveable arm. It reflects an image of the workpiece carrying table below the arm to the camera. A manipulator is also mounted at the end of the arm. The entire assembly can be rotated as a single group.

The inner workings of the camera are centered within the camera housing and are attached to a ring gear. This gear is driven by a stepper motor to rotate the image tube of the camera.

A second video camera is positioned above the work carrying table. This camera has a wider field of view which is capable of spanning the width of the table surface. A Model TN2201 camera also made by GE is suggested. This camera serves as a finder for the workpiece positioned on the table. Once a workpiece is detected by this camera, the computer controlling the device moves the arm to the general location of the piece on the table. After the arm has been moved to the general location, the narrow angle camera takes

over control.

The overall device is intended to find a randomly positioned workpiece, pick it up, and orient it to a standard position. This operation is useful for feeding workpieces to dies of a production machine, such as a punch press.

For the device to operate, the computer must first be programmed with information about what the workpiece will look like when in the proper position. This is performed by positioning a master workpiece on the carrying table in the correct orientation necessary for feeding the production machine. The narrow angle video camera mounted on the robot arm is centered on the master workpiece and focused. The camera is then rotated by the stepper motor until the image of the workpiece is properly positioned in the field of view of the camera. The proper position is defined by lining up an edge of the workpiece with the x axis of the image plane of the camera. The number of picture elements along this edge is stored in the computer along with the entire image.

In operation, a randomly arranged workpiece placed on the carrier table is first detected by the wide angle overhead camera. The computer analyzes the image and calculates the workpiece's approximate xy position. This information is used to drive the robot arm to the approximate location of the workpiece. Once the arm is in the approximate location, the workpiece is within the field of view of the narrow angle camera mounted on the arm.

The next task for the robot to perform is centering the randomly oriented workpiece within the narrow angle camera's field of view. To accomplish this centering, the computer matches the areas in the four quadrants surrounding the image of the workpiece. The quadrants are set by establishing axes that bisect the array of pixels of the workpiece image. This may be done by storing the number of occurrences of x and y coordinates in two separate arrays. These arrays constitute a frequency distribution table. It is a simple calculation to find the median points in both tables. These become the x and y axes coordinates. The computer then measures the difference in area between the quadrants and calculates how much movement is required to center the workpiece.

Once the workpiece has been centered, the computer must determine whether or not the workpiece is properly oriented. The camera picture is compared with the image of the centered master workpiece stored in memory. The camera is not rotated until the overall height is measured and compared to the height of the master. If the height is the same, the image is compared element by element. If the comparison is successful, the workpiece is in the correct orientation. If the height is not the same, the camera is rotated one increment and the overall height is again measured and compared. This process is continued until successful.

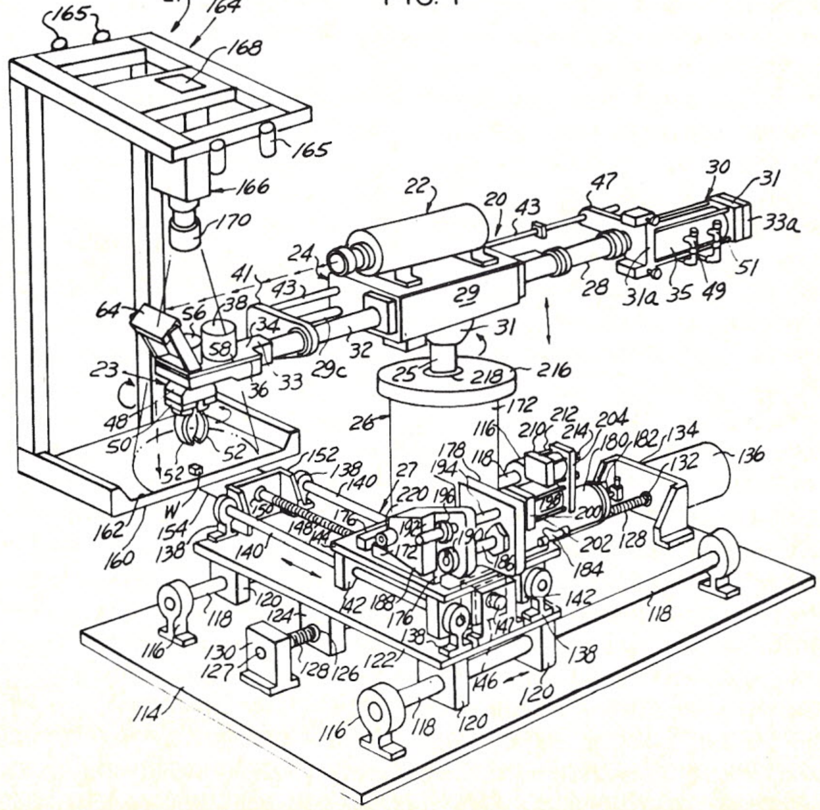
Once successful, the gripper at the end of the arm is rotated to the same relative orientation as the narrow angle camera and directed to pick up the workpiece. The piece is then positioned in the production machine. The entire operation is accomplished with a minimum of mathematical calculations.

The Kirsches' patent indicates that the entire system may be controlled by an Intel 8080-based microcomputer (specifically an IMSAI 8080). Other specific part numbers are listed for various sections of the overall robot system. The patent also refers to two other patents. Patent number 3,777,902 discloses details of the overall mechanism. Patent number 3,406,837 describes the xy axis positioner.

Copies of all three patents may be ordered directly from the U.S. Patent and Trademark Office for $.50 each. Orders should be sent with payment to: Commissioner of Patents and Trademarks, Washington, DC 20231. □

U.S. Patent Feb. 5, 1980 Sheet 1 of 11 4,187,051

FIG. 1



USE YOUR APPLE AS A ROBOTICS DEVELOPMENT SYSTEM

Paul Hollingshead
1235 Wildwood Avenue #30
Sunnyvale, California 94086

A personal computer can be a powerful tool for learning about robotics. It can analyze pictures, control motors, record readings from instruments, and send commands at specific intervals. A computer can simulate the real world, analyze circuits, translate instructions to codes the machine can understand, and help write reports on the results. Here we'll look at some of the things an Apple II computer can do in your robot lab, and some of the accessories that can help it do more. Addresses for the various vendors providing these products are listed in the accompanying text box. If you know of other hardware or software products, please write. Apple owners that read *Robotics Age* would appreciate it, and so would I.

Languages. An Apple II computer provides a range of programming capabilities to match the speed, memory, and program length requirements of many robotics applications. Five programming languages are supported by Apple Computer Inc., and many others are available from outside publishers. Assembly-language programs are often used to control special peripheral devices and develop speed critical procedures. The DOS Toolkit and the Apple (UCSD) Pascal Language System products from Apple both have assemblers to perform the translation of low-level instructions to 6502 machine code. Integer BASIC, available in read-only memory or on diskette, provides relatively rapid operation from a high-level language. Applesoft BASIC provides more useful real-number and trigonometric functions while sacrificing some speed.

For structured programming, Apple Pascal is a winning choice. Pascal provides a sensible way to create long programs that can be understood by, and shared with others. Programs written in Apple Pascal can be run by anyone who has UCSD Pascal on his system, and vice versa. Apple has recently separated the Pascal software from the memory expansion card that they used to provide. If you already have a 64K-byte Apple, Pascal can be added for half of what it cost previously. The Pascal language system also gives you access to FORTRAN, a very popular engineering language.

Game Port. Built-in to the Apple is a game port that provides a very simple interfacing capability to signals from the outside world, through which a program can read three 74LS series TTL inputs. Four outputs of a similar nature are also provided. There are connections for four variable resistors that are commonly used for game paddles, or joysticks, but could be put to other uses. Another input acts as a strobe that changes from +5 V to 0 V whenever certain addresses are read.

Peripheral Cards. The best feature of an Apple is the eight peripheral slots mounted on the back of the chassis. Dozens of manufacturers build add-on boards that plug into those slots. Boards that are especially useful for robotics experimentation are listed in the following section, and addresses for the manufacturers are provided in the accompanying text box.

A/D and D/A Converters. Temperature, position, and pressure are a few of the real-world quantities you might want a robot to sense. If the quantity you wish to measure can be represented by a varying voltage, you can use one of the many available analog-to-digital (A/D) converters that converts a DC voltage into a number that can be used by your program. California Computer Systems' 7470A A/D board measures signals from −3.999 to +3.999 V on a single channel. Conversions can be done in 400 msec. Other cards offer multiple channel capabilities to keep track of more than one signal. Applied Engineering has an inexpensive 8-channel board that has 8-bit resolution (converts a voltage into a number from 0 to 255). The company offers owners a newsletter in which to share programs. Interactive Structures sells a board that has 12-bit resolution for 16 channels. The conversion can be triggered by an external signal, as required. TecMar has a board that has 16 channels and 12-, 14-, and 16-bit accuracy. Another TecMar board converts a number to an analog signal. This board

has 12-bit accuracy and two channels.

Mountain Computer (formerly Mountain Hardware) offers both A/D and D/A conversion functions for 16 channels on a single board. Eight-bit resolution is provided over a range of plus or minus 5 V. They also provide a diskette that includes diagnostic software.

To measure a wider range of voltages, or to measure resistance and current, Sabtronics International offers their 2020 digital multimeter than can be interfaced to an Apple.

Stepper Motor Controllers. Stepper motors can be used to provide precision motion for a robotic arm, hand, or small automatic machine tool. Robotic Synergy Inc. has a pair of boards that allows an Apple II to simultaneously control three stepper motors. When plugged into an Apple slot, their Bus Expander board amplifies the signals from the Apple bus for any purpose although it was primarily intended to drive their Parallel/CY – 3 + 3 stepper motor controller. This board uses the CY512 integrated circuit to translate sophisticated commands to the signals necessary to drive the motors. It can control 4-phase motors, with on-board amplifiers to provide 5 amps for each phase. Two controller boards can be driven by one Bus Expander board, providing six axes and control from one Apple II slot. TecMar Inc. plans to go into production with a single board that can control two stepper motors.

BSR X-10. Lights, coffee pots, stereos, and other appliances that plug into a wall socket can be manipulated by an Apple II using a BSR X-10 controller. The Versacard from Prometheus includes driver circuitry for the BSR X-10 system as an option. The Apple Cat modem from Novation also has this option.

Speech Synthesis and Recognition. To recognize spoken commands and to respond with speech are important aspects of robot usefulness. The robots in Isaac Asimov's stories recognized speech before they could produce it, but speech synthesis is actually a bit ahead of recognition in today's world. Steve Ciarcia explained how to build a board, programmed with phonemes, that can have an unlimited vocabulary (September 1981 *BYTE* magazine). That hardware, with software to translate text to phonemes, is available from Micromint Inc. John Bell Engineering has a similar package. Mountain Computer's SD200 system will digitize phrases of your choice and record them, to be played back under program control. This type of speech sounds better than that generated by the phoneme synthesizers, but puts a practical limit on the vocabulary available.

For speech input, the VET/2 system from Scott Instruments recognizes 40 phrases of up to 1.5 seconds each while requiring just over 10K bytes of memory. The software included has a method of substituting the voice input for the Apple keyboard. Existing programs can then use voice input. Voicetek offers a system called Cognivox, which plugs into the Apple game port, recognizes 32 words, and

speaks a separate vocabulary of 32 additional words. These vocabularies can be stored on disk and read in appropriately.

Prototyping. A number of manufacturers have prototyping boards that allow you to build your own devices. Apple Computer Inc. offers a prototype package that contains information about the bus signals. California Computer Systems offers prototype boards for both conventional wire-wrap and soldered techniques; it also offers a board cut to size with connector fingers etched out and the rest of the board blank, so you can etch it to your own circuit design. For troubleshooting a board while it is in the chassis, the company has an extender board that raises any standard Apple II plug-in board out of the chassis while still maintaining electrical connections.

Expansion Chassis. With so many useful add-ons available for the Apple II, it isn't difficult to fill the eight slots provided in the chassis. What is more likely, however, is that the cards will consume all the power. Mountain Computer has a solution. Their expansion chassis provides an additional 30 watts of continuous power for use by eight more slots. The expansion chassis is bank selected, though language cards in slot 0 of the main chassis will stay active when the expansion chassis is chosen.

An Apple personal computer can be a versatile tool for learning about and experimenting with robotics. It can measure, control, and record information at lightning speeds, and analyze and display solutions to large problems. There are a wide variety of accessories that can be added, and with the dozens of manufacturers involved, more will be forthcoming. □

Addresses of Firms

Apple Computer Inc.
10260 Bandley Drive
Cupertino, CA 95014
(408) 996-1010

Applied Engineering
POB 470301
Dallas, TX 75247
(214) 492-2027

California Computer Systems Inc.
250 Caribbean Drive
Sunnyvale, CA 94086
(408) 734-5811

Interactive Structures Inc.
112 Bala Avenue
POB 404
Bala Cynwyd, PA 19004
(215) 667-1713

John Bell Engineering Inc.
POB 338
Redwood City, CA 94064
(415) 367-1137

Micromint Inc.
917 Midway
Woodmere, NY 11598
(516) 374-6793

Mountain Computer Inc.
300 El Pueblo Road
Scotts Valley, CA 95066
(408) 438-6650

Novation
18664 Oxnard Street
Tarzana, CA 91356
(213) 996-5060

Prometheus Products Inc.
42577 Fremont Blvd.
Fremont, CA 94538
(415) 490-2370

Robotic Synergy Inc.
1336 South 1100 East
Suite 201
Salt Lake City, UT 84105
(801) 485-3365

Sabtronics International Inc.
5709 North 50th Street
Tampa, FL 33610
(813) 623-2631

Scott Instruments
1111 Willow Springs Drive
Denton, TX 76201
(817) 387-9514

TecMar Inc.
23600 Mercantile Road
Cleveland, OH 44122
(216) 464-7410

Voicetek
P.O. Box 388
Goleta, CA 93116
(805) 685-1854

IBM ROBOTS

Photographs and figures courtesy of IBM.

James F. Derry
1538 Rowles Drive
Akron, Ohio 44313

IBM has lifted the veil of secrecy surrounding its robotics program and revealed a measure of the research and development it has been doing for the past decade. What has been revealed for all to see is a system that gives meaning to the phrase "Intelligent Machine":

A pair of robots (the IBM 7535 and IBM RS-1 Manufacturing Systems) suited for high volume, light assembly, test, and precision fabrication work;

A commitment to software and communications serving robotic manufacturing systems;

Computer-based control sub-systems;

"A Manufacturing Language" (referred to as AML) suitable for programming robots;

Educational courses covering these systems and their maintenance.

Some observers have concluded that IBM is going after the entire CAD/CAM industry, while others point out that selling the IBM Personal Computer with the 7535 is a great way to sell computers. A more likely target is market share in the overall international robotic/automated factory systems business.

IBM appears to consider automated assembly as a difficult challenge in robotics and of considerable importance to the future of its business. In keeping with IBM's announced concept of aiming at the software and communications aspects of robotics, both robots contain digital I/O capabilities. Indeed, the crown jewel of these systems appears to be the AML language. After all, they are referred to as manufacturing systems and they are exactly that.

These robots are now being tested in both IBM and customer plants. Almost every IBM plant has at least one robot in service. Design optimization of these robots for high-volume light assembly work is in keeping with IBM manufacturing needs. Hence, if you need spot-welders, paint-sprayers, or heavy lifters, you will need to talk to someone else about the heavy iron, but IBM may be interested in the robot control systems and in-plant communications.

IBM 7535. The smaller of the two robots, the IBM 7535 Manufacturing System, is a pick-and-place machine (see photo 1). Although it is manufactured by Sankyo Seiki (Japan), it is built to IBM specifications. It consists of a pedestal-mounted table-top unit able to swing its arm (using electric motors) through 200 degrees, a wrist attached to the end of the arm able to swing through an arc of 160 degrees, and a gripper attached to the end of the wrist and aimed downward which is able to rotate (roll axis) 180 degrees in either direction as shown in figure 1. The arm may be moved up or down a total of 3.9 inches by an air cylinder; the gripper field of action envelope begins 7.33 inches from the pedestal center-line and extends to 18.27 inches from the center-line; maximum payload is 13.2 pounds including the gripper and its tooling. Both the gripper and the table top are furnished by the customer.

Major features of the 7535 include:

Cartesian coordinates, point-to-point
Multiple point positioning under microprocessor control
Position repeatability plus or minus 0.002 inch
Speeds up to 57 in/sec
Automatic acceleration selection (using a cam curve)
Detachable programming unit (the IBM Personal Computer)
Easily altered control program.

The unit could be used in many operations typical of the automotive, appliance, electronic, and cosmetics industries, such as:

Automatic assembly
Parts handling
Multiple-point drilling, chamfering, tapping
Assembling and inspecting many parts in-process
Packing parts into cases or removing them
Machine loading and unloading.

The on-board control unit is a microprocessor capable of storing up to five programs (sequential, multi-point operations) in its 6K-byte memory. Selecting a program is simply a matter of selecting a switch setting.

The IBM Personal Computer is used as a programming unit for this robot, and a subset of the AML language, AML/Entry, resides in the computer for this task. The computer is attached to the controller during programming and may then be detached for use with another 7535 unit or any other use the programmer may put it to. The Personal Computer can be used on-line with the 7535. A teaching pendant is also available for use in teaching the robot sequences of movements and operations.

The cost of the IBM 7535 alone is \$28,500. The Personal Computer configured for this application costs \$4575 and the AML/Entry program adds another \$1000. With this total of

$34,075, a selection of grippers and associated tooling plus training may run a budget up to $50,000. Deliveries are to begin in the fourth quarter of 1982.

This robot can be very useful as an educational tool for manufacturing engineers. The cost is reasonable, it can perform many useful functions in a factory environment, it uses a high-level manufacturing-oriented language, and the IBM Personal Computer is a very useful and versatile device by itself.

IBM RS-1. The RS-1 (shown in photo 2) is a much more sophisticated robot than the 7535, providing a larger work envelope, six degrees of freedom for gripper motion versus the 7535's four, an IBM-furnished gripper equipped with strain gauges for tactile sensing, the ability to use optical sensors, and a full-blown IBM Series/1 computer as its controller. Figures 2 and 3 show the various degrees of freedom available to the RS-1.

The work-space of the RS-1 is more obviously defined since the arm is mounted in a rigid rectangular frame that includes a waist-high work bench as a work surface. The arm is mounted from the top of the frame so that it hangs down toward the work surface; movement is limited to 18 inches in the x direction, 58 inches in the y direction, and 17 inches in the vertical z direction. Top speed is 40 inches per second and repeatability of positioning is 0.008 inch.

The IBM-furnished manipulator attached to the bottom of the arm with wrist swiveling devices can move through 270 degrees on the roll axis (a vertical axis), 180 degrees on the pitch axis, and 270 degrees on the yaw axis. Each of these movements can be done at speeds up to 180 degrees per second.

The maximum arm payload is 5 pounds, not including the gripper weight. Hydraulic power is used for arm motions and positons are controlled with servo feedback loops. A hydraulic linear motor is used (smaller in size than a human fist), which has a very high torque to weight ratio.

Tactile sensors for the gripper are strain gauges mounted on the tip, side, and pinch surfaces of the gripper fingers. Photo-detectors can also be

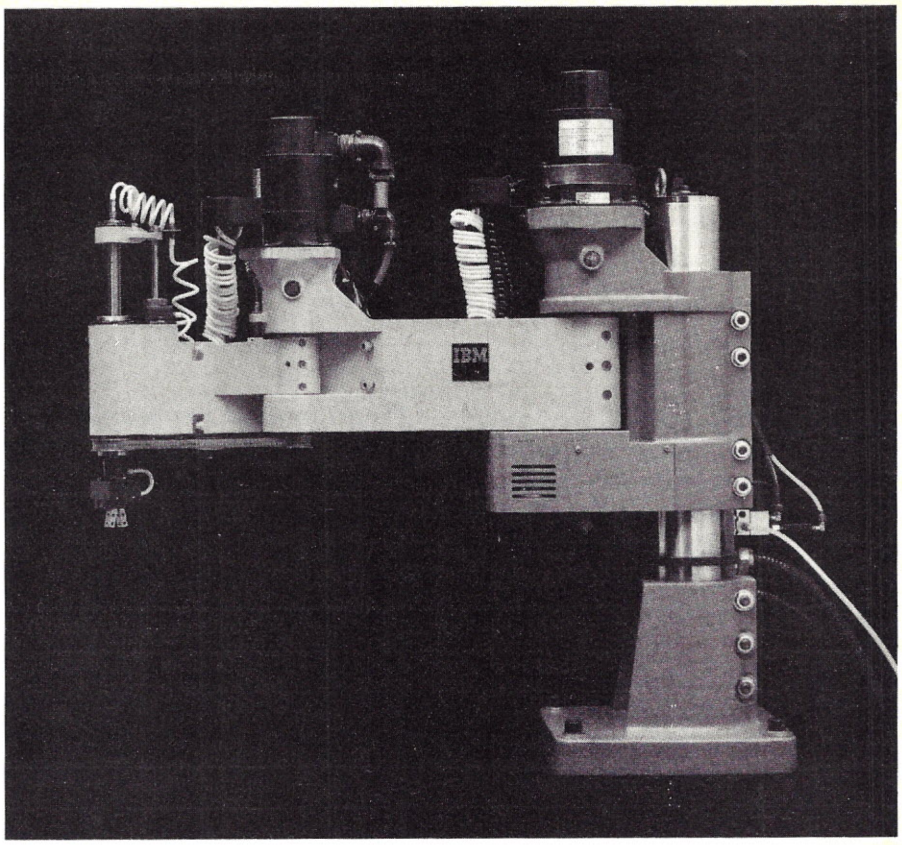


Photo 1: The IBM 7535 Manufacturing System arm.

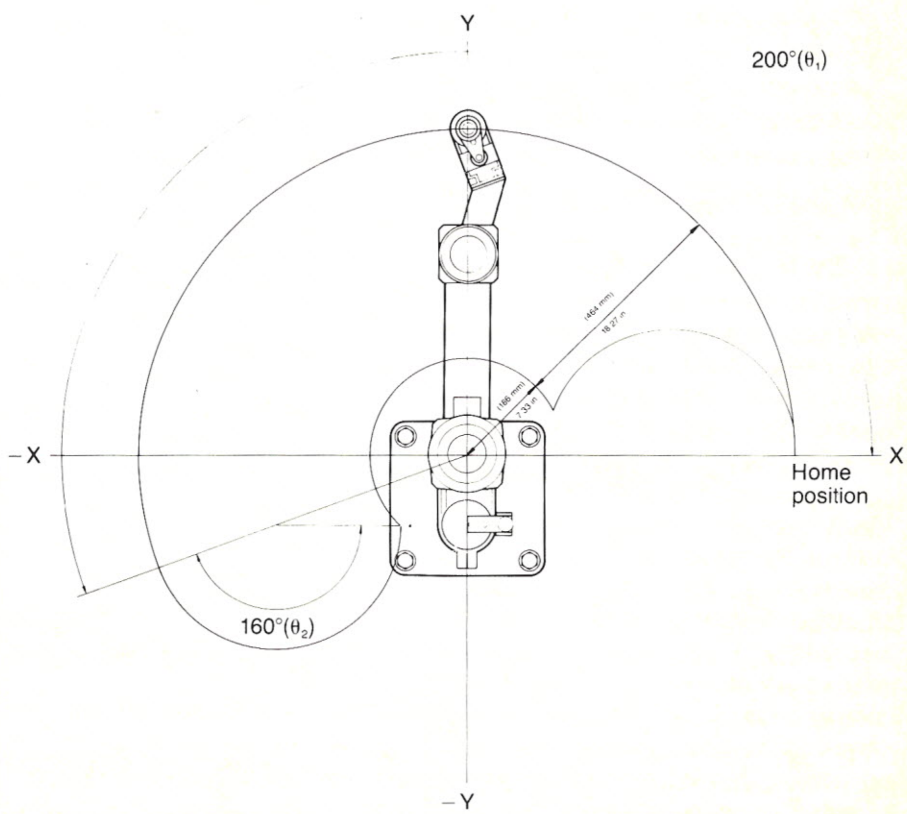


Figure 1: IBM 7535 Manufacturing System workspace and arm-wrist motions. Limited z-axis motion is available through the up and down motion of an air-powered piston.

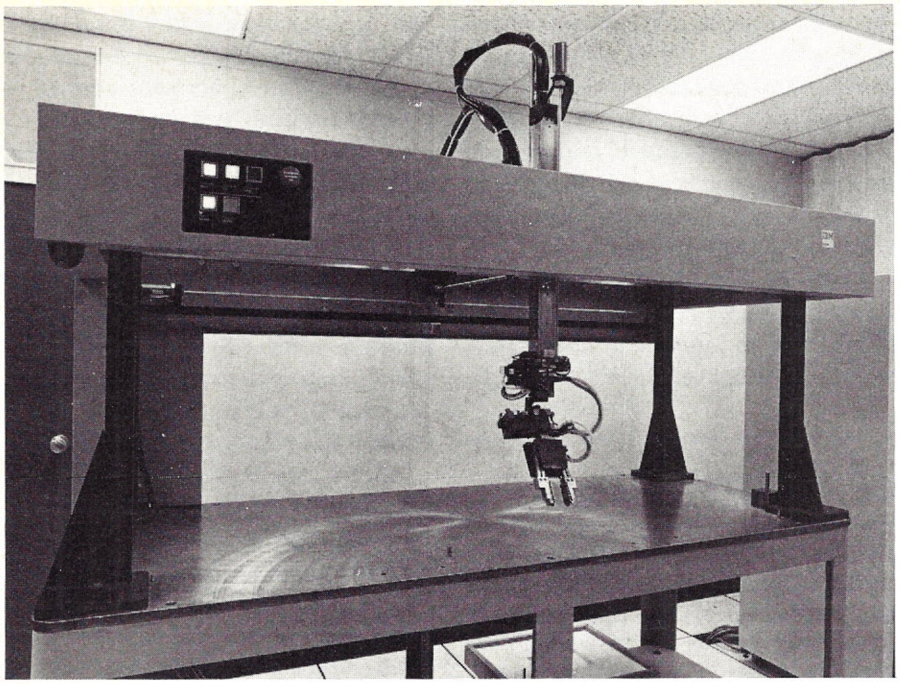
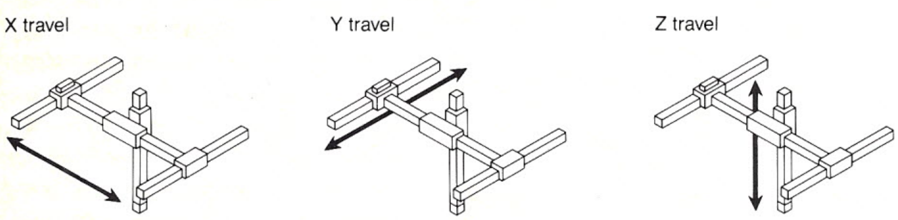
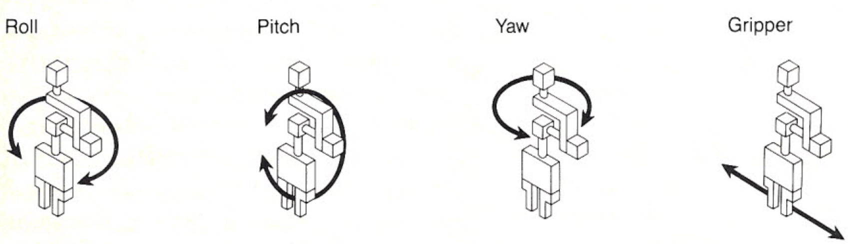


Photo 2: The IBM RS-1 Manufacturing System showing the RS-1 robot arm and the Series/1 computer.



Axis	Travel	Top speed
X	18 in. (457 mm)	40 in./sec (1,016 mm/sec)
Y	58 in. (1,473 mm)	40 in./sec (1,016 mm/sec)
Z	17 in. (431 mm)	40 in./sec (1,016 mm/sec)

Figure 2: IBM RS-1 Manufacturing System arm motions.



Axis	Travel	Top speed
Roll	270°	180°/sec
Pitch	180°	180°/sec
Yaw	270°	180°/sec
Gripper	0-3.25 in. (82 mm)	5 in./sec (127 mm/sec)

Figure 3: IBM RS-1 Manufacturing System wrist and gripper motions.

used to allow non-contact positioning.

The operator's control panel is located at the top left of the frame; panel pushbuttons are back-lighted and color-coded for start-up, motion, and emergency off.

Key features of the RS-1 include:

Adaptibility, flexibility
Stored application programs with fast change over
Computer control
Programming pendant (when needed)
User sensors may be attached.

Flexibility is a primary advantage, allowing fast change overs to new tasks and relocations on the shop floor. The RS-1 system may be moved to a new location, a new program may be called into main memory, and the system be set to performing a new task in a matter of minutes or hours rather than the several days or weeks required for more traditional automated assembly methods.

The RS-1 allows dynamic workspace calibration. When a new work cart with parts for assembly is brought to the robot, the robot arm touches a reference post on the cart. The gripper fingers sense the actual position of the cart and the computer then adjusts position and motion data in the program accordingly.

The computer polls robot operations every 20 ms. If anything is faulty, the robot is automatically shut down. The RS-1 sensors can also detect an empty feed box condition, signal the operator, and proceed to other tasks until the feed box is replenished.

The RS-1 has been designed for light assembly, fabrication, testing, material handling, electronic parts insertion, and other intricate work. IBM is currently using the RS-1 to assemble word processor machines (Displaywriter, Austin plant), to insert gears and ribbons into typer cartridges, and for placement of eighty different character slugs into printers.

RS-1 systems are in service at fifteen customer sites, and twenty-five additional RS-1 systems are being added to the service test. The RS-1 Manufacturing System is not yet being offered for

sale and prices are not available; the introduction date may occur in 1983.

AML. IBM's "A Manufacturing Language" appears to be the key to its robotic systems. Communications and a systems approach to robotics design form the central issue; ability to tie into manufacturing data bases and computer-aided-design equipment is considered critical.

Although robots can easily be led through sequences of motions and operations manually in a "teaching" mode using a pendant control unit, that method is too inflexible for the IBM concept. Pendant control and manual teaching is not sufficiently adaptable to overcome even minor problems that arise on the shop floor.

AML contains features of various languages, including APL, PL/I, and LISP. It is an interactive interpreter, reentrant, recursive, and able to handle real, integer, and string data types. AML is intended to provide both robot and peripheral equipment control in order to establish complete control over an entire work environment. It also allows frequent changes to the work environment, permitting adjustments for misfed parts and other out-of-the-ordinary situations. An engineer/programmer developing a robotic control program using AML can observe the robot's responses to misaligned, misfit, misoriented parts or other maladies and develop algorithms to cope accordingly.

AML's capabilities allow two robots, e.g., an IBM 7535 and an IBM RS-1, to function together. The 7535 could be used to unload parts and pass them on to the RS-1 for assembly; the 7535 might then accept the finished work from the RS-1 for subsequent routing or packaging.

Commands in the AML language have been chosen so that the language can be quickly learned and effectively used by operators with limited programming experience. Commands are intuitively familiar to the operator because they mean what they say in plain English: APPROACH, WITHDRAW, TRANSPORT, GRASP. Two hundred such instructions are available.

AML was aimed at the intermediate volume mechanical assembly production range, lying between the very high volume applications using dedicated "hard" automation and the low volume applications using hand tools and manual labor. Many different problems have been encountered for programming within this environment. The three-dimensional world of assembly is spatially complex and hard to visualize. The required attention to detail is greater than that for scientific or business computing. Control of intricate mechanisms with real-time requirements is new to many programmers. Techniques that are well defined in digital computer programs are often imprecise in the real world of robots.

Be that as it may, IBM has a videotape that it has screened publicly. T. G. Donlan (see *Barron's*, 5 April 82, pages 8, 9, 20) reports his reactions:

> *RS-1, the IBM robot system, is much more than a mechanical arm;*

The IBM Personal Computer

IBM has designed its personal computer around the Intel 8088 processor. This gives the IBM Personal Computer a 16-bit word size (and full code compatibility with the Intel 8086) while using an 8-bit data communication bus. Addressable memory runs up to 1M byte, though physical memory is currently limited to somewhat smaller sizes. A 40K-byte ROM contains necessary system routines.

Standard diskette drives are 5¼ inch, can hold 160K bytes, are single-sided, and two may be installed in the standard configuration. RS-232C interfaces (9600 bps maximum) are used for communications using an asynchronous protocol. Computer system diagnostics include memory parity checking and power-on self-testing.

The standard system printer uses a 9 by 9 character matrix and prints at speeds up to 80 cps, is bidirectional, can have up to 132 characters per line. It currently supports twelve font styles.

Companion products in the Intel line are numerous and widely used: the 8087 numeric data coprocessor, 8089 I/O processor, a megabit bubble memory, Ethernet boards, and many other devices. Products looming over the horizon include a database back-end microcomputer, operating systems in firmware (software in silicon), local area network devices, and faster forms of the 8086.

It is no surprise that several vendors of Intel-based products have moved into the IBM Personal Computer market with add-on products. One such vendor, TECMAR Inc. (Cleveland, Ohio), jumped in with both feet by announcing the PC-MATE, an expansion chassis that matches the IBM Personal Computer decor. TECMAR has announced twenty-five PC-MATE products including Winchester disks, analog/digital converters, digital/analog converters, serial and parallel I/O interfaces, memory, system clock with battery back-up, EEPROM, BSR X-10 controller, stepping motor controller, video image digitizer, IEEE-488 Multibus interface, prototyping board, music synthesizer, voice synthesizer, extender board, and dust cover.

A new start-up vendor, Xedex Corporation, has introduced a Z80 board that plugs into the IBM Personal Computer chassis. This board, Baby Blue by name, enables the IBM Personal Computer to run CP/M-80 programs without modification. However, the 5¼-inch diskettes furnished with the IBM are not guaranteed to be CP/M standard and individuals may encounter problems when transporting the CP/M programs to the Personal Computer.

As I write this article, the first software support packages for the Personal Computer have begun to appear. Sorcim is offering a new version of its SuperCalc spreadsheet program and SuperWriter word-processor program. Lifeboat Associates has created an Intel 8086 library that will run on the Intel 8088 under the IBM Personal Computer disk operating system. Software Products International Inc. has a series of interactive multiprogram business packages written in UCSD Pascal which use a relational database management system.

When the IBM Personal Computer is attached to the IBM 7535 controller for robotics programming, the add-on options may or may not be useful. However, the Personal Computer is likely to be detached once the 7535 is running properly and there is no point in letting such a fine, versatile computer gather dust in a corner. It can then be used as an engineering computer, a manufacturing resource, a software development system, or whatever else may be appropriate and useful.

it is a complex computer system for design, testing and production. It also includes an advanced three-dimensional dynamic modeling system, which allows an engineer to draw and manipulate diagrams of solid objects on the television screen of a computer terminal. . . .

The most important thing is that the system is connected to the robot. Computer modeling makes it possible for an operator to create an animated three-dimensional drawing on a television screen, showing a pair of robot arms using interchangeable tools to assemble electronic devices out of small parts. The animation is automatically translated into the program that drives the real robot, which makes real devices. . . .

Dynamic animation, as described by Donlan, is unfortunately still in research and unavailable.

IBM Series/1. The IBM Series/1 computer been used for some time as a process control, factory automation, and communications machine. It can handle digital and analog I/O as well as non-IBM devices and protocols.

The dedicated IBM Series/1 computer, when used as a controller for the IBM RS-1 Manufacturing System, is intended to have a full set of peripherals. Attached devices include a 120 cps matrix printer, a video display terminal with full keyboard, hard disk and floppy diskette drives, servo circuits for control of the manipulator arm, and ports for attaching the digital I/O devices.

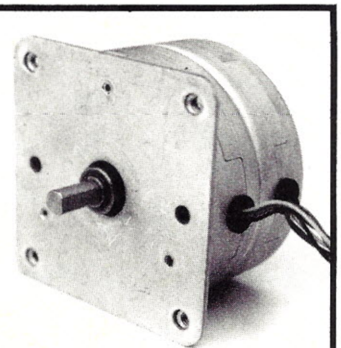
Software for the Series/1 includes AML. Data-processing support routines may also be added if desired for system monitoring, data acquisition, management reporting, and other tasks. A programming pendant is provided which can be used to "teach" the robot arm sequential tasks; all pendant switches, push buttons, and indicators are software programmable.

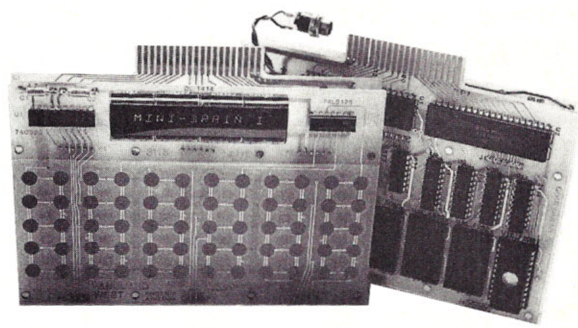
Education. IBM is offering several robotics education courses in Boca Raton, Florida.

Course R-01, Robotic Executive Concepts, one day, is offered on an invitation-only basis. The course covers the concepts of flexible automation, the role of IBM robotic systems in the automated factory, the IBM 7535 Manufacturing System, IBM internal robotic applications, financial justification for robots, development of implementation plans, overview of robotic programming, IBM offerings in support and services, and resources needed to implement a robotic application.

Course R-02, IBM 7535 Concepts, Analysis, Programming, and Implementation, is a $400, two-day session. It covers topics similar to those presented in R-01, but in greater depth, and it is aimed more specifically toward the 7535 system: introduction to the 7535, application, economic considerations, programming, materials handling and parts feeding concepts, tooling and layout considerations, product design for assembly, and selection and development of applications.

Course R-03, IBM RS-1 Robotic Ap-





Calendar

Intermexpo Show

More than 40 manufacturers and distributors of machine tools and related metal working equipment will exhibit in the first INTERMEXPO Guadalajara Tool and Manufacturing Engineering Conference and Exposition to be held in Guadalajara's Forum Tapatio July 7-10, 1982.

Sponsored by the Society of Manufacturing Engineers, Dearborn, Michigan, and its Guadalajara Chapter No. 562, the four-day event also will offer a two-day seminar, three one-day workshops and technical sessions at the nearby Camino Real Hotel.

A wide range of equipment will be demonstrated including machine tools of all types, lathes, drills, presses, welding equipment, instruments and gages, and inspection, testing and safety equipment. Companies from the U.S.A., Mexico, Italy, Japan, Taiwan and other countries will exhibit.

Exposition hours are 12:00 to 20:00 hours daily (12 noon to 8:00 p.m.), Wednesday through Saturday, July 7-10.

The seminar and workshops announced for INTERMEXPO Guadalajara are:

July 6-7, CNC Math and Programming, conducted by F. J. Cosio-Marron, President, CMEI, Ltd., Inc., Columbus, Ohio.

July 7, Robots in Modern Manufacturing, conducted by William R. Tanner, CMfgE, President, Productivity Systems, Inc., Farmington, Michigan, U.S.A.

July 8, Designing Low-Cost Jugs and Fixtures, conducted by Edward G. Hoffman, CMfgE, Consultant, Colorado Springs, Colorado, U.S.A.

July 9, Hand and Wave Soldering, conducted by Joseph Sylvester, President, Omni Training Corp., Covina, California, U.S.A.

To obtain complete conference information by mail or telephone, contact the following persons: Gary Bachman, Administrator/Special Programs, Society of Manufacturing Engineers, One SME Drive, P.O. Box 930, Dearborn, Michigan 48128, U.S.A. (Telephone 313/271-1500, Ext. 375), or Esperanza Ayon, Etoile Organizacion de Congresos, Fuente dela Cibeles No. 9, Tecamachalco, Mexico 10, D.F. (Telephone in Mexico City, 2-94-08-30).

SME Slates New IP Event for Long Beach (Ca.), Oct. 6-8

The Society of Manufacturing Engineers (SME) will hold its first Long Beach Industrial Productivity Conference and Exposition at the Long Beach (CA) Convention Center, October 6-8, 1982.

Machine tools and industrial equipment for all phases of plant operations will be highlighted along with conference sessions focusing on new manufacturing technologies and management techniques.

Exposition hours at the Long Beach Convention Center will be 12:00 Noon to 8:00 p.m. on Wednesday and Thursday, October 6 and 7, and 11:00 a.m. to 5:00 p.m. on Friday, October 8.

The concurrent technical conference will examine several key manufacturing topics including industrial robots, CAD/CAM, tooling for aerospace, layout planning, composites, and hand and wave soldering.

For additional information about the Long Beach Industrial Productivity Conference and Exposition, contact the Public Relations Department, Society of Manufacturing Engineers, One SME Drive, P.O. Box 930, Dearborn, Michigan 48128, Telephone 313/271-0777.

Robot Research, Developments and Applications in Canada

A conference jointly sponsored by the Central Ontario Chapter of Robotics International Society of Manufacturing Engineers and National Research Council of Canada.

This conference will be held at Delta Inn, Mississiauga (Toronto), Ont. on September 20, 21, 1982.

A variety of technical papers and presentations include topics on robot research and developents, applications, controllers, programming languages, sensory feedback, education and training.

Further details of the conference can be obtained from: RI-SME Conference Secretariat, 6535 Mississauga Road, Mississauga, Ont. Canada, L5N 1A6.

READER SERVICE

Product #	Company
1	Ackerman Digital, p. 5
2	AMSI, p.27
3	BEI Electronics, p. 20
4	CSI, p.10
5	Datacube, Cover III
6	Datametrics, p. 11
7	E/Z Associates, p. 7
8	Gledhill Electronics, p. 28
9	Intellimac, Cover IV
10	Micro Systems Science, p. 23
11	Microbot, p. 17
12	Moog Valve, p. 2
*	North American Technology, p. 35
14	PSI, p. 11
15	Robotics Industry Directory, p. 27
16	Robotics Age, p. 35
17	Sandhu Machine Design, Cover II
18	Space Studies Institute, p. 39
*	Tech-0, p. 33
20	Vanguard West, p. 33
21	Vision Peripherals, p. 39
22	Western Electric, p. 1

*correspond directly with company

plication System Design, runs three days and costs $600. Besides covering topics from the courses already described concerning automated factories, the 7535 system, economics, and applications, there are periods devoted to hands-on use of the RS-1 robot in a teach-by-example (pendant) mode, types of programmable robot systems, internal IBM applications, selection of robotic applications, justifying a robotic system, feeding and escapement of parts, and programming the RS-1.

Course R-04, IBM RS-1 Application Analysis and AML Programming, is the full five-day stint at $1600. It covers much additional information: RS-1 specifications, safety procedures, overview of AML, data definition and manipulation, motion command controls, disk organization and utilities, program development tools, the interactive environment, subroutine structure, logic control expressions, sensor/monitor commands, data management, error recovery/operator interface, debugging aids, and a selection of advanced topics.

IBM Series/1

The IBM Series/1 constitutes a complete family of 16-bit minicomputers that can be configured in the general price range of $10,000 to $100,000. The design intent was to furnish a line of versatile, inexpensive computers that could perform a wide variety of functions: sensor-based applications, data processing, communications, or distributed data processing.

Many of the Series/1 machines have been used by the communications common carriers for back-room functions such as directory assistance, toll-call recording, and replacement of electromechanical components. Recently IBM has installed its telephone store-and-forward package on the Series/1 machines.

Four communication line disciplines are supported: SDLC (synchronous data link control), BSC (binary synchronous communications), ACC (asynchronous communications control), and synchronous. Other protocols may be used if desired. The Series/1 can be used as a terminal or device network controller and can be interconnected in any of several ways to form distributed networks in system-to-system or system-to-host configuration.

An interrupt capability using an 8-bit word provides automatic vectoring to any of 256 direct interrupt entry points. The interrupt structure is elaborate and designed to permit rapid context switching using hardware versus extensive software.

A Processor I/O Channel provides an asynchronous, multidrop interface that can service up to 256 devices differing in speed and technology. It supports direct program control and cycle-stealing modes of operation as well as interrupt servicing.

The Series/1 is a flexible workhorse that can be used in many environments. Hence, its use as a controller for the RS-1 robot is an excellent choice.

There are also courses in maintenance of the IBM 7535 and the IBM RS-1.

More detailed information may be obtained from the IBM Corporation, Advanced Manufacturing Systems, 1000 NW 51st Street, Boca Raton, FL 33423, (305) 998-2000. □

ADAPTING A SPEECH SYNTHESIZER

Mark Robillard

3 Peach Lane

Townsend, Massachusetts 01469

Voice output from computers has long been desirable, mainly because of its natural machine-to-human communication application. In recent years a number of speech-synthesis products for consumers have appeared on department store shelves, and not long ago the first voice output peripheral became available to the personal computer owner. One of these, the Radio Shack TRS-80 Voice Synthesizer, manufactured by Votrax, uses phoneme reconstruction to actually synthesize human speech. Being an unlimited vocabulary machine, the Votrax unit can be used in output applications where standard predefined vocabularies from other synthesizer vendors do not exist. One such specialized field is robotics.

While performing extensive research on robot communications, I found it necessary to interface this synthesizer to the TRS-80 Color Computer. Unfortunately, the Model I and the color do not share the same interface. The TRS-80 Color Computer has a unique expansion connector. Both connectors being different, I searched for the universal alternative: RS-232C serial I/O.

Serial Interface. This article will show you how to interface the TRS-80 speech synthesizer to any serial port, and will provide details for an interface and vocabulary builder program for the Color Computer. First, let's take a look at the synthesizer as it comes when you purchase it.

Figure 1 is a partial schematic — that is, it shows the portion of the board that is discernable. The rest is either sealed or unmarked. The 74S133 is a 13-input NAND gate which is wired as an address decoder. It is looking for an address of decimal 16352 or higher. With

Table 1: Relationship between Votrax phoneme sounds and ASCII characters used to produce them.

VOTRAX	ASCII	DECIMAL	VOTRAX	ASCII	DECIMAL
PA1*	SPACE	32	A1	@	64
I2	!	33	AH2	A	65
I	"	34	B	B	66
I3	#	35	CH	C	67
OO	$	36	D	D	68
OO1	%	37	E1	E	69
Y	&	38	F	F	70
U	'	39	G	G	71
IU	(	40	H	H	72
A2	)	41	I1	I	73
AY	*	42	J	J	74
NG	+	43	K	K	75
AW	,	44	L	L	76
0 DEC.	–	45	M	M	77
E	.	46	N	N	78
ER	/	47	01	O	79
PA0*	0	48	P	P	80
AW1	1	49	DT	Q	81
AW2	2	50	R	R	82
EH1	3	51	S	S	83
EH2	4	52	T	T	84
EH3	5	53	U1	U	85
UH1	6	54	V	V	86
UH2	7	55	W	W	87
UH3	8	56	ZH	X	88
AE1	9	57	Y1	Y	89
AE	:	58	Z	Z	90
AH1	;	59	O2	[	91
THV	v	60	O	/	92
TH	=	61	AH	]	93
SH	v	62	A	v	94
SELECT	?	63	NULL	—	95

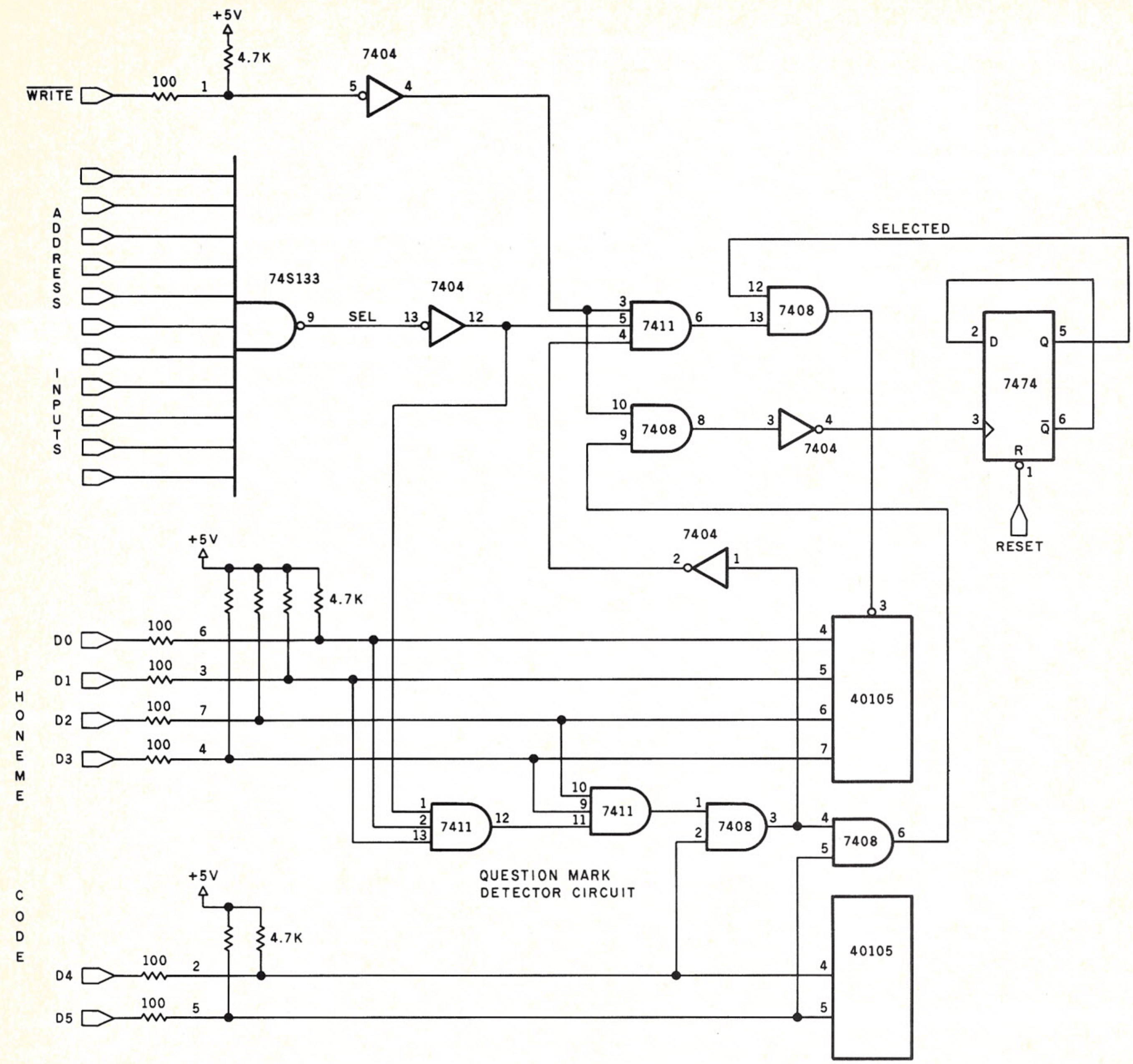


Figure 1: Partial schematic of Votrax speech synthesizer board. This diagram shows only the entry portion of the speech circuit.

the serial interface we have no use for this gate. Examining further, we see two 40105 ICs that are FIFO (first-in/first-out) registers that store 32 phoneme codes. The entire writing circuit is gated on and off by the 7474 shown in the upper right. This flip-flop is triggered on when a question mark is received, gated off when another one arrives. What you see in this partial schematic is all you need to see.

Construction Details. Open the back of the unit. Remove the volume knob and press in the select LED. The circuit board can now be removed from the rear. It will only go as far as the speaker cable will allow. Perform the following modifications to the board:

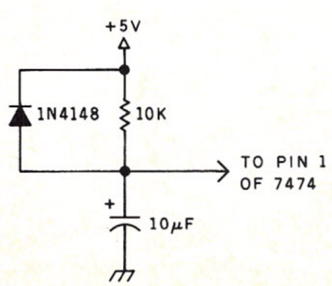


Figure 2: Power-up reset circuit to be added to 7474.

Remove the 40-pin ribbon cable.
Cut pin 9 of 74S133.
Add circuit to pin 1 of 7474 (figure 2).
Cut slot in rear panel for serial cable (photo 2).

This is all the modification necessary to the basic unit. Now on to the new circuit. Figure 3 depicts the serial interface. The 6402 is a CMOS UART (universal asynchronous receiver/transmitter) and is widely available. The 75189, 4069, and 8046 are also readily available. The received data output lines go to pins marked on the

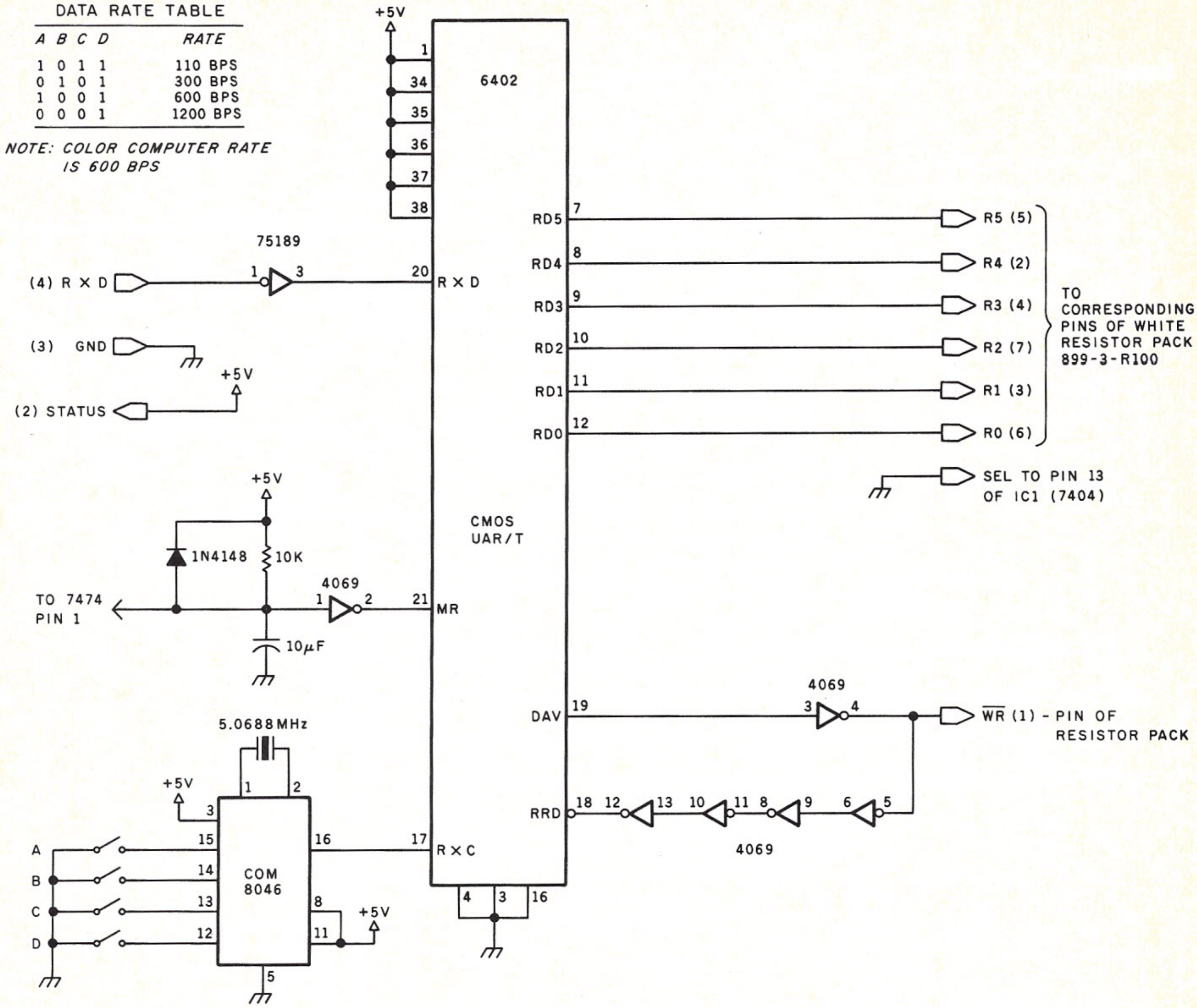


Figure 3: Serial interface to speech synthesizer.

resistor pack located beside the ribbon connector. Five volt power and ground can be picked up at any TTL chip on the board.

Using the Synthesizer. The Votrax synthesizer uses units of sound referred to as phonemes. These pieces of words form the basis of every known word in the English language. With this type of synthesis, large amounts of words can be stored in a computer with relatively little memory. Table 1 lists the 64 phoneme sounds available with the synthesizer and the ASCII characters used to represent them. This type of speech unit has been well documented in the January/February issue of *Robotics Age*, as well as in the March/April article devoted to Type-n-Talk. Additional articles are listed in the references at the end of this article.

Interfacing to the TRS-80 Color Computer cartridge port would require a special circuit mounted in a box, and

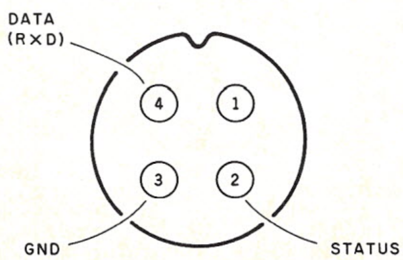


Figure 4: TRS-80 Color Computer serial port connections for printer.

unique poke assignments that would not be as easy to use as a print statement. After all, we are basically replacing the printed output with a voice.

The TRS-80 Color Computer has a serial RS-232C printer port built into the standard unit. The pin connections for this port are shown in figure 4. Hook up the serial port to the synthesizer and let's write some software.

This configuration allows us to talk to the synthesizer with simple Print to line-printer commands (PRINT #-2). Remember that a "?" turns the unit on and then another turns it off. After sending out phoneme strings by hand for a while you will wish you had some sort of editor.

Photo 1: TRS-80 Color Computer voice development station. Voice synthesizer rests on top of computer. A serial interface connects the two.

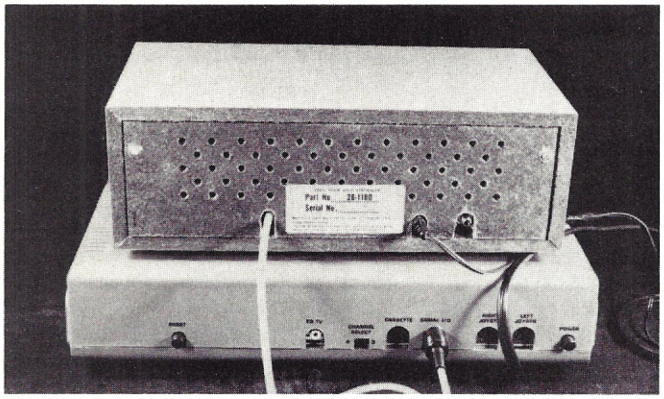


Photo 2: Rear view of synthesizer showing cutout for new cable on left.

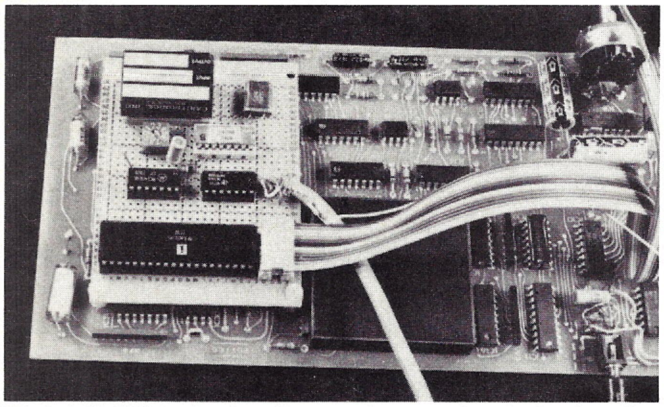


Photo 3: Serial interface board mounted on top of synthesizer board.

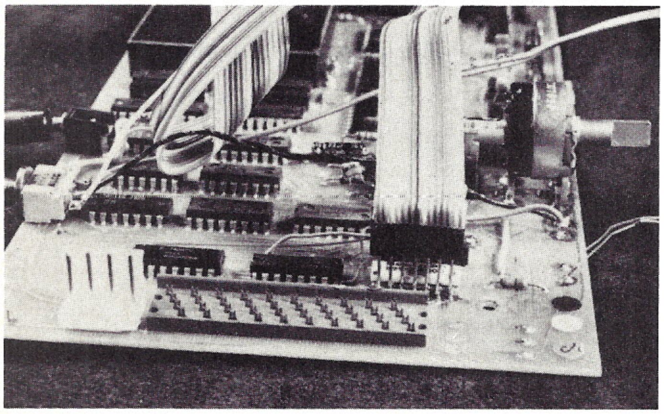


Photo 4: Connections to the resistor pack at ribbon cable connector. These wires go to the serial interface.

TALKEDIT Vocabulary Editor. Listing 1 is a program that will help develop a vocabulary with this machine. Although it is not a foolproof editor, it does provide the basics. The program provides four choices: INPUT, SAY, EDIT, and LIST. The INPUT routine lets you build a word out of characters much like typing a letter. Pressing "Enter" will send it to the synthesizer. The SAY command will speak the word currently in the buffer (last entered). EDIT will allow you to change any part of the word, much like the EDIT routine in Color BASIC. LIST will print the phoneme string on the screen.

I have utilized voice output in the program: after all if you've got it. . . flaunt it. The listing should be self-explanatory. Some enhancements could be made to provide on-line recording of finished words, or a text-to-phoneme conversion routine could be added.

Although this synthesizer is rather dated compared to currently available devices (including one from Votrax), it gives experimenters a chance to explore phoneme use while waiting for the newer parts to arrive in the mail. □

REFERENCES

Fons, Kathryn and Tim Gargagliano. "The TRS-80 Speaks," *BYTE*, October 1979, page 113.

Fons, Kathryn and Tim Gargagliano. "Articulate Automata," *BYTE*, February 1981, page 164.

Fons, Kathryn and Tim Gargagliano. "A Votrax Vocabulary," *BYTE*, June 1981, page 384.

Ciarcia, Steve. "Build an Unlimited Vocabulary Speech Synthesizer," *BYTE*, September 1981, page 38.

Listing 1: TALKEDIT voice construction editor written for TRS-80 Color Computer.

```
1 '   TALKEDIT
2 '   VOICE DEVELOPMENT PROGRAM
3 '
4 '
6     CLS(3)
8     DIM B$(50)
10    PRINT#-2, "? ?"
11    GOTO 400
15    GOTO 500
18    CLS
20    PRINT@5, "iNPUT";:PRINT@11,"sAY";:PRINT@15,
      "eDIT";:PRINT@20,"lIST";
30    A$=INKEY$
40    IF A$=" " THEN 30
50    IF A$="L" THEN 540
60    IF A$="I" THEN 140
70    IF A$="E" THEN 220
74 '
75 '      SAY ROUTINE
76 '
80    PRINT #-2, "? ";
90    FOR Y=1 TO X-1
100   PRINT#-2,B$(Y);
110   NEXT Y
120   PRINT#-2," ?"
130   FOR Y=256 TO 319
134   PRINT@Y, " "
136   NEXT Y
138   GOTO 285
139 '
140 '     INPUT ROUTINE
141 '
145   PRINT#-2,"? I#NP$T ?":X=1
146   CLS :PRINT@5,"input";
150   A$=INKEY$
160   IF A$=" " THEN 150
170   PRINT@255+X,A$
190   IF A$=CHR$(13)THEN 80
200   B$(X)=A$
210   X=X+1:GOTO 150
219 '
220 '     PRINT ROUTINE
221 '
222   PRINT#-2,"? 3DIT ?":CLS:PRINT@5,"edit";
223   FOR Y=1 TO X-1:PRINT@225+Y,B$(Y):NEXT Y
225   Z=0
230   A$=INKEY$
240   IF A$=" "THEN 230
250   IF A$=" "THEN Z=Z+1:PRINT@287+Z,B$(Z):GOTO 230
260   IF A$=CHR$(13) THEN 272
262   IF A$=CHR$(9) THEN A$=" "
270   B$(Z)=A$:PRINT@287+Z,A$:GOTO 230
272   Z=Z+1
273   IF Z >X THEN X=Z
275   GOTO 80
280   PRINT#-2, "? LI#ST ?"
281   CLS:PRINT@5,"list";
285   FOR Y=1 TO X-1
290   PRINT@255+Y,B$(Y)
300   NEXT Y
310   GOTO 20
400   PRINT#-2,"? < #ISS0!Z0 <EE&0K7L/0K6MPY(UT/ ?"
402   PRINT@107,"TALKEDIT";
403   PRINT@173,"VOICE";
404   PRINT@234,"DEVELOPMENT";
405   FOR T=1 TO 500: NEXT T
410   PRINT#-2,"? VO85&SS0D#V44L6PM43NT0SIST3M ?"
420   FOR T=1 TO 2000:NEXT T
430   GOTO 500
500   PRINT#-2,"? 3NT/033N&0768VV  < EE&Z0K6MM99NDS ?
510   GOTO 18
539 '
540 '     LIST ROUTINE
541 '
545   PRINT#-2,"? LI#ST ?"
550   CLS:PRINT@5,"list";
555   C=3
560   FOR Y=1 TO X-1
570   PRINT @255+Y,B$(Y)
580   A=ASC(B$(Y))
590   PRINT@285+C,HEX$(A);",";
595   C=C+3
600   NEXT Y
610   GOTO 20
```

CONSTRUCTING AN INTELLIGENT MOBILE PLATFORM

Part I: Fundamental Control Mechanics

Mark Robillard
3 Peach Lane
Townsend, Massachusetts 01469

Most popular transportation methods involve the use of wheels. We use them in cars, trucks, trains, and bicycles. Anyone who has survived the ordeal of teaching a child how to ride a two wheeler will agree that it's the way the wheels are positioned that spells success or dismal failure.

Vehicles use a variety of wheel placement arrangements. Some use only two wheels, others as many as eighteen. The bicycle is designed with the wheels in front of each other. This configuration requires that the payload (you) maintain balance for the vehicle. On the other hand, a car, utilizing four wheels, or two sets of bicycle-like arrangements, does not require the structure to maintain balance.

Some applications require that robotic machines move about an area in much the same manner as a car or bicycle. The same considerations must be made for the structure and wheel arrangements. Many different configurations have been constructed and all have advantages and disadvantages. However, when it comes to position control and simplicity of design it is hard to beat the two drive wheeled or treaded arrangement of the tank.

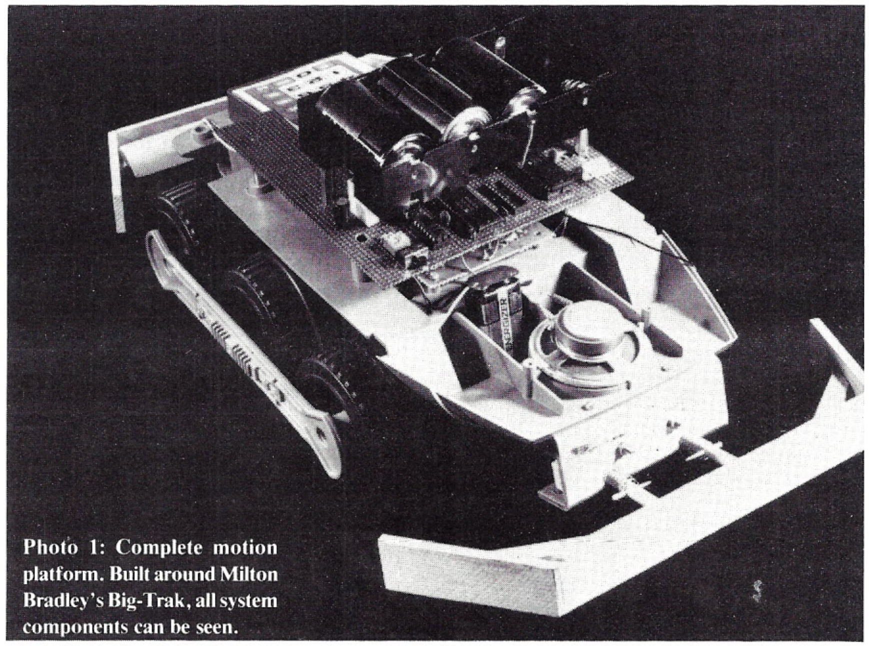


Photo 1: Complete motion platform. Built around Milton Bradley's Big-Trak, all system components can be seen.

Two Wheels are Better than One. Looking at figure 1, it can be seen that a two-wheeled vehicle with both wheels opposite each other can satisfy all the requirements for positional control without the need for a separate steering motor. Let's consider each example. In figures 1a and 1b you can see that driving both wheels in the same direction will yield forward or reverse motion. This concept is fairly standard except for the requirement of a motor for each wheel. One advantage in this mode is that the weight is evenly distributed between two motors, thus more pulling power. Of course, the real reason for having separate motors is illustrated in figures 1c and 1d. Here it can be seen that driving the motors in opposite directions will rotate the vehicle to either the right or left. Note that no forward or reverse motion is required for turns, so it is possible to "turn on a dime," so to speak, which can be very desirable. Proportional control of each motor will yield very fine turning angles.

Let's take the motor control knowledge gained in the last issue of *Robotics Age* ("An Inexpensive Arm-Hand System," May/June 1982, page 20), and apply it here. Figure 2 depicts the same two-wheeled arrangement and the required motors. Notice that because each motor is facing the opposite direction you cannot simply tie the negative or black wires together and the red or positive wires together to get the motors to rotate in the same direction. In fact, for both wheels to move in the same direction, the two motors have to be connected up the reverse of each other. Figure 3 shows one way you can connect both motors to move the vehicle in any of the four directions. Build it up with some scrap wood as the base, model car wheels, and slot car motors. Become familiar with the type of control necessary to position the vehicle properly. While you're doing that let's give it some muscle.

Gears, Gears and More Gears. Assuming that you will want this moving platform to actually carry something (either the control circuitry for the rest of the robot or maybe an arm and power supply), you will find out very quickly that slot car motors are only good for one thing.

It is necessary to "gear down" the speed of the motor, which will increase its pulling power or torque. Gearing arrangements are designed many different ways; one such configuration is illustrated in figure 4. Looks complicated, doesn't it? Yet each gear in the gear train has a distinct purpose. This configuration results in a one pound torque shaft geared down from a very small slot car motor.

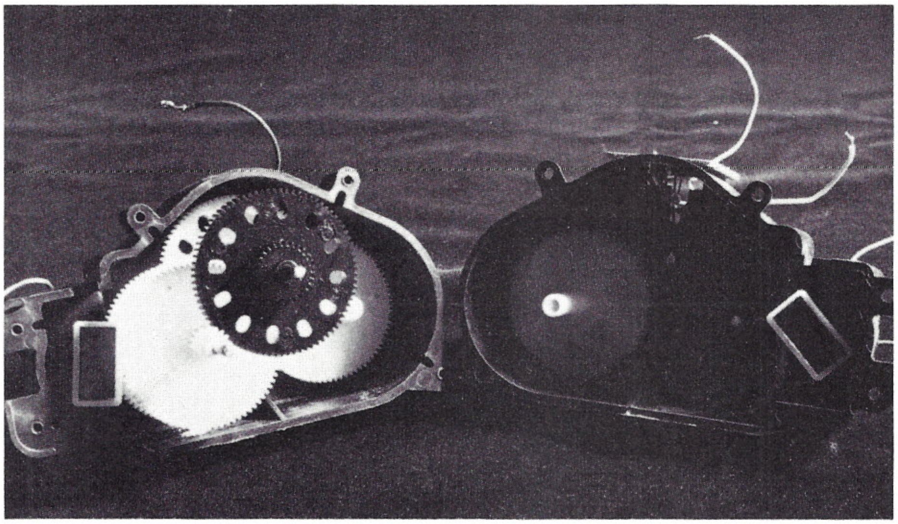


Photo 2: Open view of dual motor gear box. Top gear, in black, is used as a window for infrared sensors to sense position holes.

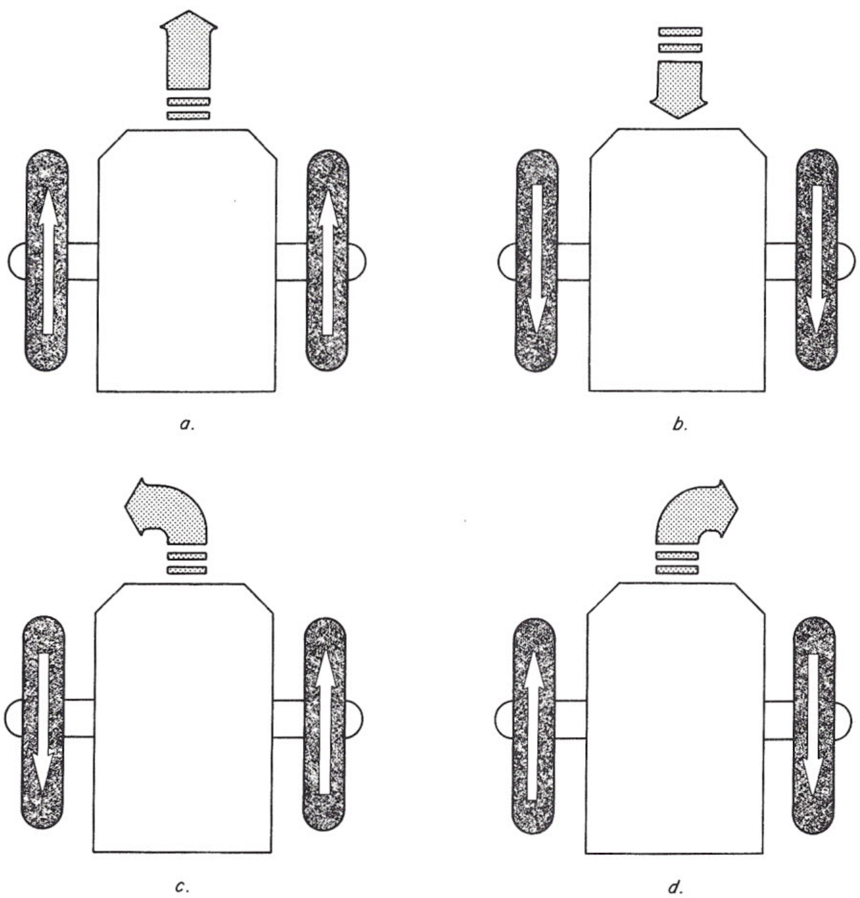


Figure 1: Motion mechanics of two-wheeled vehicle. Notice how opposite rotations of wheels turn the body.

Photo 2 shows a dual motorized gear box I picked up at a local surplus supply house. It contains all the gears shown in the figure as well as two DC motors with a magnetic clutch and two large ¼-inch drive shafts to mount wheels on. The box is made of sturdy plastic and the motor wires are terminated in quick connect connectors for easy hookup. There is also a bonus included in the package. Mounted on top of the box is an infrared light sensor and emitter board that can be used to provide feedback in later designs. The entire motor control system described in the last article in one package for under $5!

Sounds like just the thing to build a motion platform around. We will need some sort of base, wheels, other dummy wheels or casters to balance the assembly, and, of course, a battery holder for power to the motors. Also needed will be some sort of electronic interface to the motors and the photo-position sensor. When faced with a dilemma such as this, do as I do.

Back to the Toy Store. Located among the action toys you will find the now semi-famous Milton Bradley toy, Big-Trak. Toy indeed! Remember that gear box, it was designed for this toy. Also, this toy includes a single IC microcomputer, drivers for both motors, the base, wheels, a speaker with sound effects, a pulsing lamp output that can be used for other things, and a repertoire of eight motion instructions which, when punched into the microcomputer's keyboard located on the back of the unit, can position this machine just about anywhere you could want, at the touch of a button. Presto — programmable motion for less than $40!

Let's look at the instructions. Figure 5 is the layout of the keypad used for input.

This is the *forward* command. It is followed with a number from 1 to 99. This number represents Big-Trak lengths. A *forward 2* command will send the machine

straight ahead about twice the length of the unit or approximately 28 inches.

This button commands the unit to move in the *reverse* or backward direction the number of lengths specified.

The *turn right* command directs the system to spin right a specified number of units. The unit number approximations are shown in figure 5.

The *turn left* button is similar to *turn right* except the system spins left the specified amount of units.

HOLD *Hold* is a time delay button. You can wait from one-tenth of a second to 9.9 seconds. This command allows the system to settle after a fast turn before executing another.

FIRE *Fire* pulses a lamp on the front of the unit. You can program up to 99 pulses at a time. Later I'll show you how to use this output to provide a TTL-level signal for use elsewhere.

OUT The *out* button is used with the optional dump truck accessory which attaches to a two-conductor plug on top of the unit. Basically, it turns on the output for approximately four seconds, then off. I'll also show how to use this command to provide a "command done" signal.

RPT This is the *repeat* function. It allows you to create a loop in the program by specifying how many previous instructions to repeat.

GO *Go* starts Big-Trak performing the instructions you have programmed.

CLR *Clr* erases any program stored in memory.

You can input up to sixteen instructions into Big-Trak's memory before hitting the *go* command. The memory will remember these commands as long as power is applied. Subsequent pushes of the *go* button will repeat the actions programmed.

Having all this capability already built into a commercially available unit makes motion platform design very easy. Of course, in order to have a com-

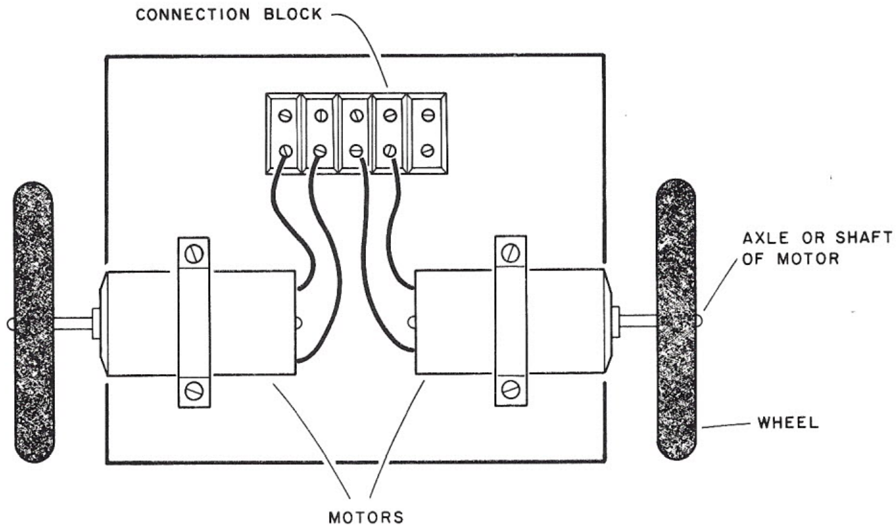
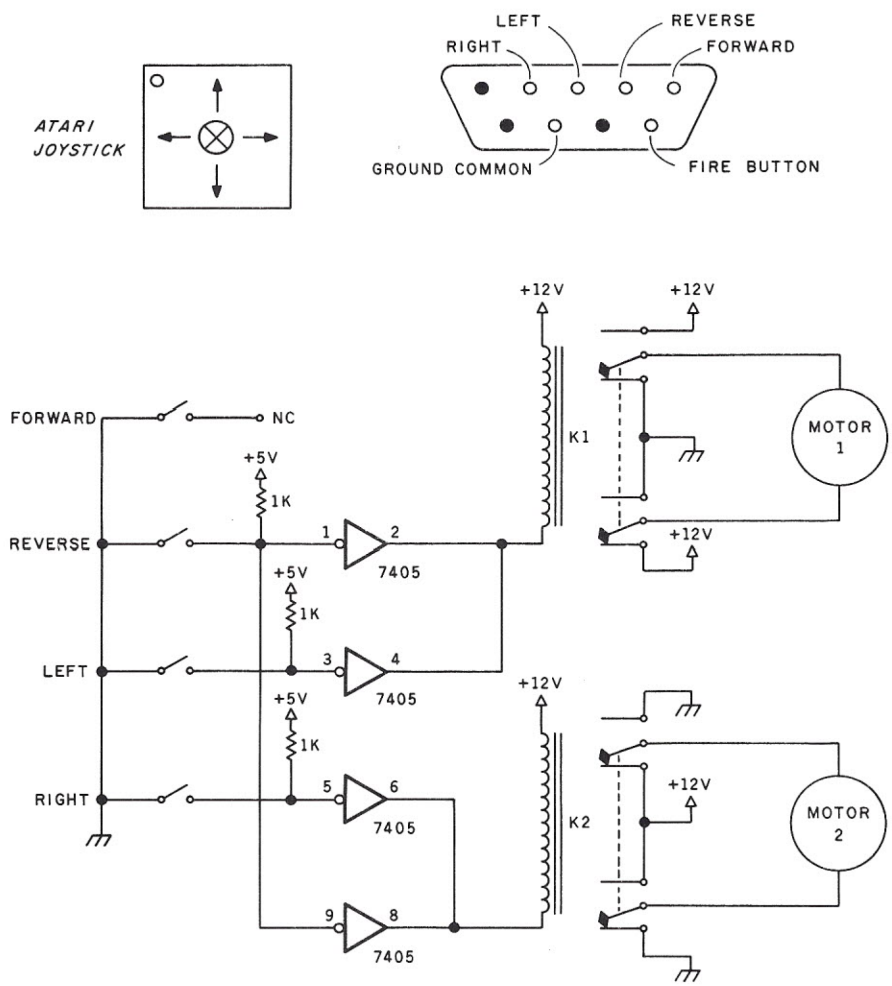


Figure 2: Two-wheeled body implementation using two motors. In this configuration the motor shafts are connected directly to the wheels.



1 PKG.	MOTORS	RADIO SHACK # 273-208
2	RELAYS	RADIO SHACK # 275-213
1	7405	ANY SURPLUS STORE
1	BREADBOARD	RADIO SHACK # 276-158
1	JOYSTICK	ATARI JOYSTICK

(Four switches could be substituted for joystick)

Figure 3: Four-direction control of two-wheeled vehicle, utilizing relays and an Atari video-game joystick.

pletely interactive system, some modifications must be made. The first job is allowing an external electronic controller or computer to command Big-Trak.

Big-Trak's Computer. Figure 6 is a partial schematic diagram of Big-Trak's controller circuit board. The heart of the system is a single-chip microcomputer manufactured by Texas Instruments. The IC scans the keypad, remembers commands, executes commands, controls the motors through appropriate driver transistors, and checks the position of each wheel. Not bad for 28 pins. All you do is hook up a 9 V battery and throw the power switch.

This controller works on the principles outlined in my last article. Each instruction is executed, then the next is fetched from the control program memory. Of course, in this particular system, the control program, I/O, and

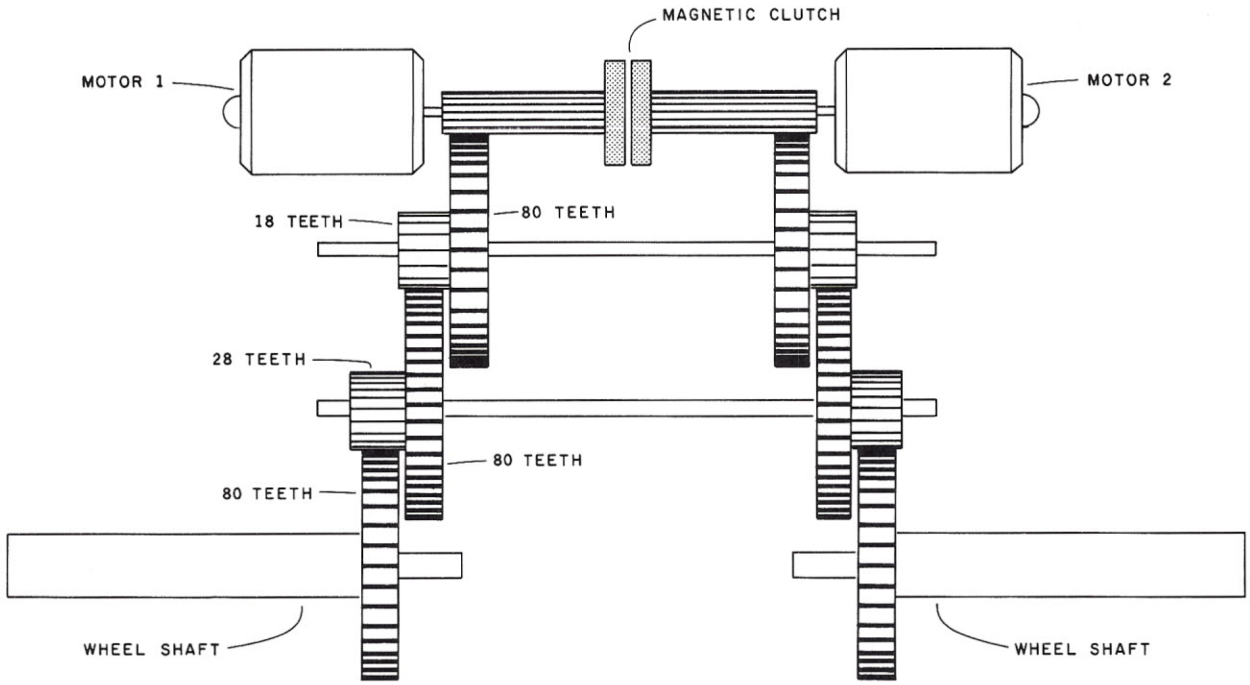


Figure 4: Wheel gearing arrangement of commercially available Big-Trak gear box.

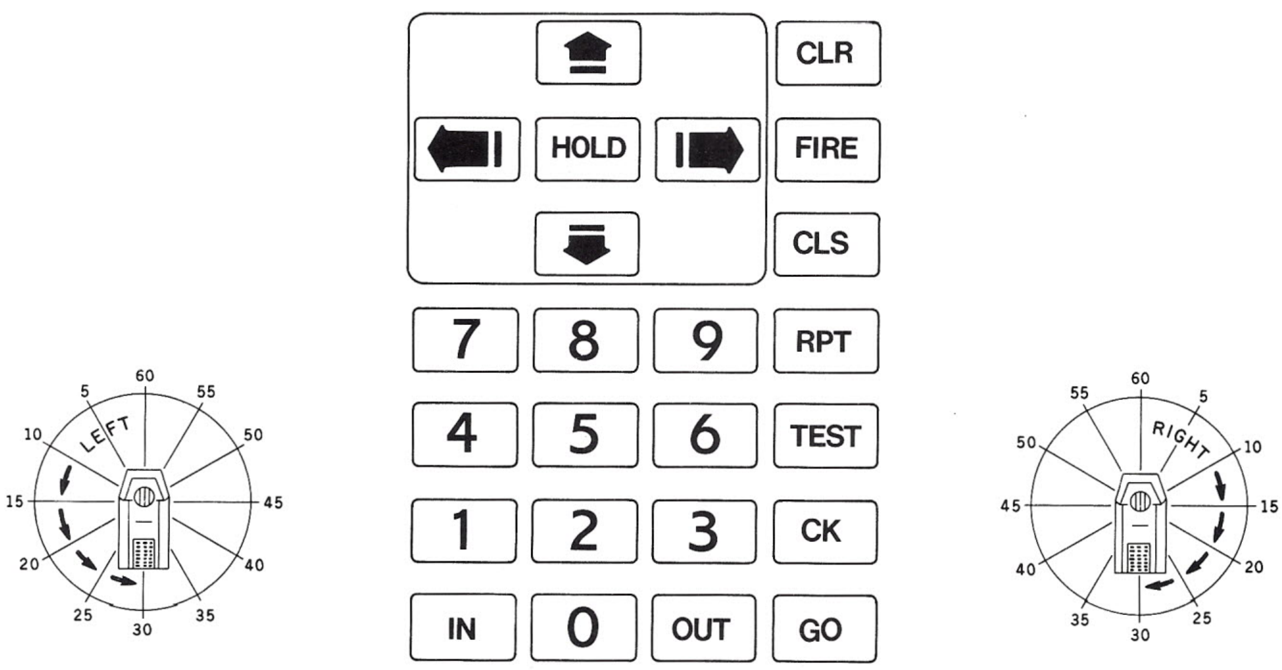


Figure 5: Keyboard of Big-Trak. To the right and left are compass points for use as a guide when keying in *right turn* and *left turn* length units.

processor are resident in the same physical part. Although this means we do not have access to the control program, interconnection is easier for the designer. A keypress signals the processor to perform a motion command which might be made up of several hundred machine control instructions. When no command is being entered and the unit appears to be dormant, it is, in reality, constantly scanning the keyboard, awaiting the touch of your finger.

The keypad is made of a sheet of mylar plastic with conductive paint on one side arranged as switch contact pads. Pressure on a pad area shorts two pads together, closing the path. This type of keyboard technology should be familiar to everybody now that it's being used in all types of consumer products.

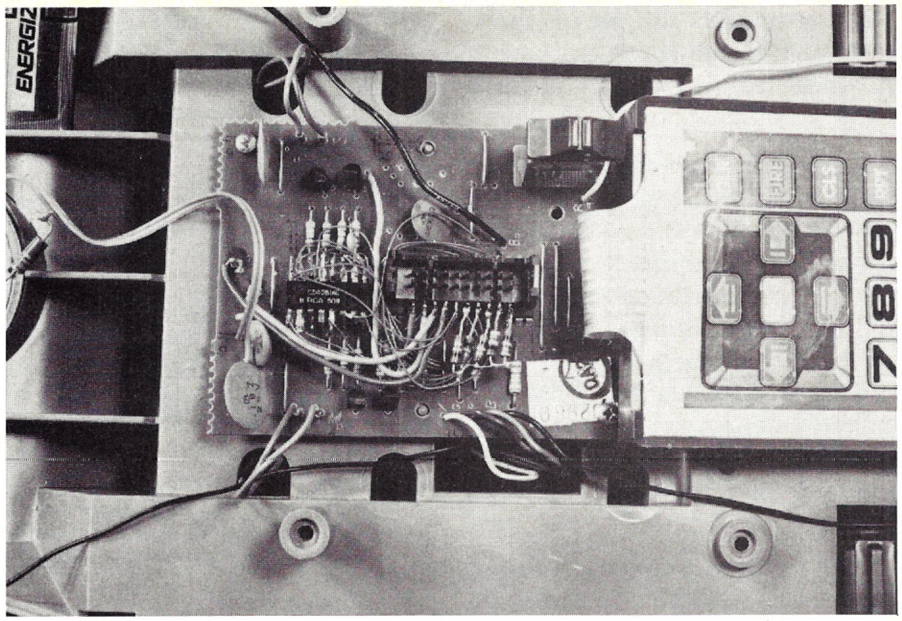


Photo 3: View of interface wired directly onto existing IC on Big-Trak controller board. This interface allows emulation of control keypad by external computer.

External Interface. The controller electronics in Big-Trak run with a 9 V power source. Normal TTL ICs won't work with such a high power level, so you will have to use CMOS technology as that type of logic can withstand up to 18 V. When choosing a connection point into a commercial circuit, you should consider the implications of how the additional circuitry might affect the circuit operation. Some systems are designed so critically that one load or connection to a certain point may cause the unit to fail. I chose to connect into Big-Trak's least critical design point, the keypad. Emulating a key function is relatively simple with off-the-shelf standard CMOS circuits.

Before we go off with our soldering irons, let's decide what this interface must accomplish. First, the commands listed earlier will have to be input through this circuit: the numbers 0 through 9 and all key presses. What about some feedback from Big-Trak? We will need to know when Big-Trak is finished doing what we commanded because its design does not allow key presses during program execution. Without feedback our external programmer would command a non-listening machine.

Thus far, the following commands and inputs will be required of this interface:

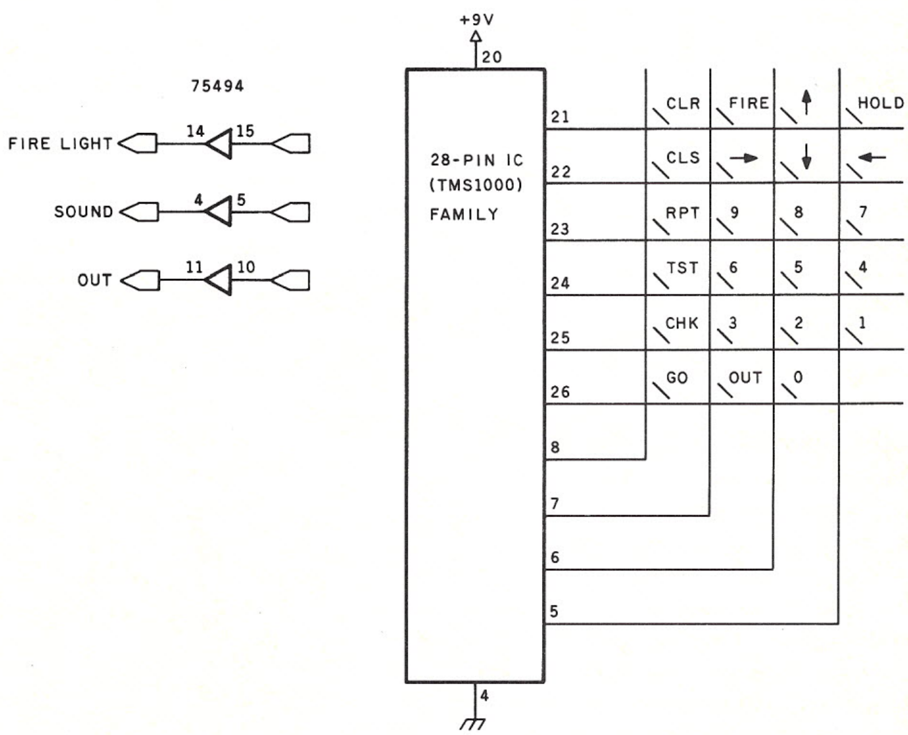


Figure 6: Big-Trak controller partial schematic diagram. This board is mounted in the center of the unit. Covers must be removed for access.

FORWARD	- output	OUT	- output
REVERSE	- output	REPEAT	- output
RIGHT TURN	- output	GO	- output
LEFT TURN	- output	CLR	- output
HOLD	- output	0-9	- output
FIRE	- output	DONE	- input

These commands will be emulating keypresses, therefore they will have to be asserted, then promptly removed much like a finger press.

Figure 7a is a complete schematic of the interface. I built mine on top of the existing ICs on the board to conserve space. You will notice the absence of any "done" interface circuitry. That will come later. For now, just re-route the wires from the lamp and the *out* jack, as indicated on the schematic. I mounted a 20-pin connector to the circuit for connection to an external computer. Photo 3 shows the completed interface.

With the entire circuit consisting of only two CD4051 ICs, I think it appropriate to describe what these do. The CD4051 is an 8-channel analog switch. An analog switch acts rather like a relay — it will pass variable signal levels accurately, whereas a digital switch or gate will only pass a 1 or 0. This type of switch is necessary in this application since the keyboard scanning signals expect to pass through an actual switch. The analog switch approximates the action of an actual switch better than any other component. Figure 7b shows the electrical switch connections inside the CD4051 circuits as connected to the controller board. Notice the similarity between this diagram and that of the keyboard circuit in the controller schematic of figure 6.

The big difference between the keypad and the analog switch circuit is the way it is actuated. The five inputs to the circuit labeled D0 to D4 are digital signal lines for connection to the external computer. The signal level of these inputs should be around 9 V, therefore the computer we connect here must use an interface circuit to convert standard TTL-level signals to their 9 V equivalents.

These signal lines have been arranged to allow a 5-bit code of 1s and 0s to activate functions. The codes I have chosen according to design are shown in table 1. Notice there is a distinct code for each function button. This code is implemented with a 1 equaling a 9 V level and, of course, a 0 representing a normal ground to 0.8 V signal.

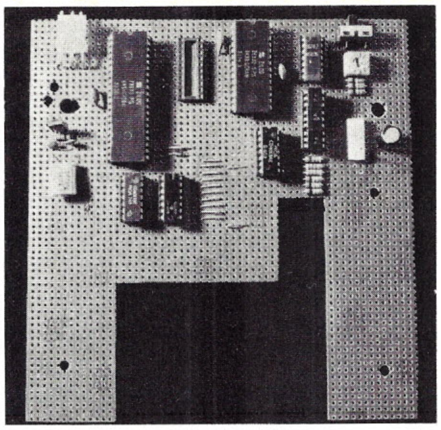


Photo 4: Local control board built using Z8. This board mounts directly over the existing controller board. A 3M-type connector is mounted on the bottom of the board so that when it is placed on top of the other board, the two are connected.

Computer Remote Control. Anticipating that you may want to connect the motion platform to an external computer for checkout, I have performed some software design to help in this interconnection. Listing 1 is written in standard Microsoft BASIC, and it contains a package of subroutines useful to the motion-control programmer. I chose BASIC because it is relatively easy to translate into other languages and readily available to the experimenter. (Also, I already know we can program the planned control in BASIC.)

Let's examine the subroutines. Robot motion planning can require sophisticated mathematical evaluations. You will probably want to use a BASIC with floating-point capability, and motion commands should not detract from the language. These routines are called much like any subroutine and therefore do not require modification to the machine.

To call a command simply write the following:

C$ = "FORWARD, 12":GOSUB 60000

If you want a command other than *forward*, just substitute it, and the same goes for the number of lengths. In fact, you can specify lengths as a standard variable such as:

C$ = "HOLD, X":GOSUB 60000

As long as the variable is a single letter, it is allowed. If you wish to relocate the subroutine to any place other than 30000, you can simply substitute your subroutine start address after the GOSUB. Either a comma or space may separate the command from the variable. You can place these com-

COMMAND WORD	CODE D4	D3	D2	D1	D0	DECIMAL EQUIVALENT
FORWARD	1	0	0	0	1	17
REVERSE	1	0	0	1	0	18
RIGHT	0	1	0	1	0	10
LEFT	1	1	0	1	0	26
CLR	0	0	0	0	1	01
HOLD	1	1	0	0	1	25
REPEAT	0	0	0	1	1	03
FIRE	0	1	0	0	1	09
OUT	0	1	1	1	0	14
GO	0	0	1	1	0	06
0	1	0	1	1	0	22
1	1	1	1	0	1	29
2	1	0	1	0	1	21
3	0	1	1	0	1	13
4	1	1	1	0	0	28
5	1	0	1	0	0	20
6	0	1	1	0	0	12
7	1	1	0	1	1	17
8	1	0	0	1	1	19
9	0	1	0	1	1	11

Table 1: The five input lines D0 through D4 shown in figure 7 are used to generate the commands shown.

mands in line with standard BASIC statements as long as you avoid the following variables in your program.

NC	TA	D$
DR	VA	CW$
DA	DC	V$

The listing is commented as to function, and should be fairly easy to understand. The interface will have to connect to your system somehow. The computer I used was a TRS-80 Color Computer with a parallel interface plugged into the cartridge port. The POKE commands do the actual transmitting of code words. Modify this part according to your system requirements.

The last portion of the subroutine checks the status of DONE and something called BUMP. We will discuss these signals later. For now, you can put a program patch in line 60733:

```
60733 RETURN
```

This will short out the DONE check but will not implement the clear function after the command. For the moment it will be necessary for you to push the CLR button between each command. Otherwise, the commands entered will all be stored in memory, and the second GO command will execute both the command entered and the last command entered.

To use the subroutine package to transmit commands for input into BigTrak's memory, simply remove the *out* instruction transmit line, 60720, and the *go* command line, 60730. This can be done by placing a GOTO 60750 command at line 60720. When using the program in this manner it will be necessary for you to transmit the GO command when you are ready for motion to begin. The use of the *out* command will be discussed in the next section.

Next Issue: *Now that the basic interface electronics are designed and constructed, the next major task is adding another on-board computer to control BigTrak's actions. Part 2, Local Computer Control, describes the implementation of a control system using a very simple single-chip computer design and the associated control software.*

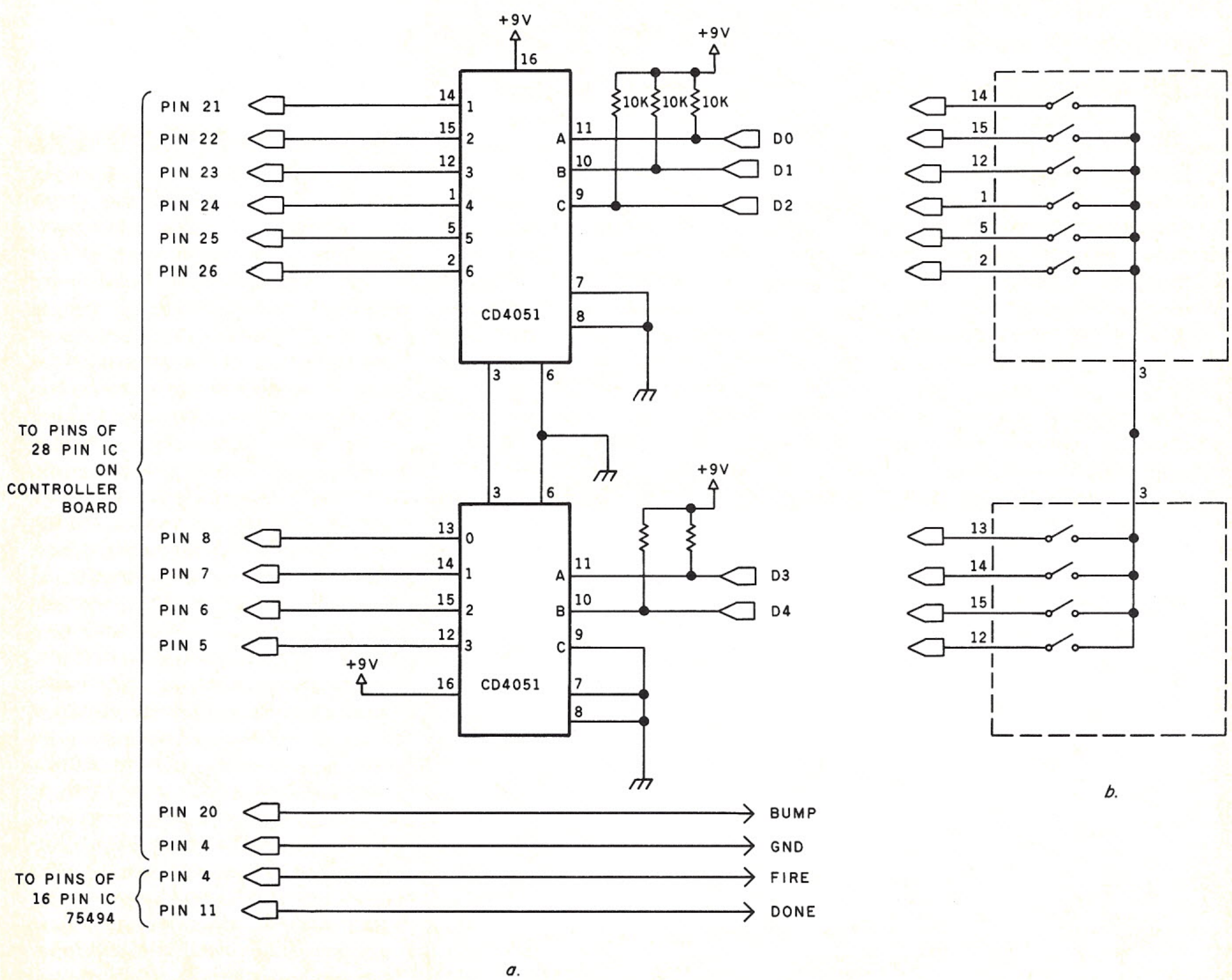


Figure 7a: External interface using CD4051 Analog switches. Figure 7b shows schematic equivalent of analog switches.

New Products

Lease, Don't Buy, The $6 Million Man

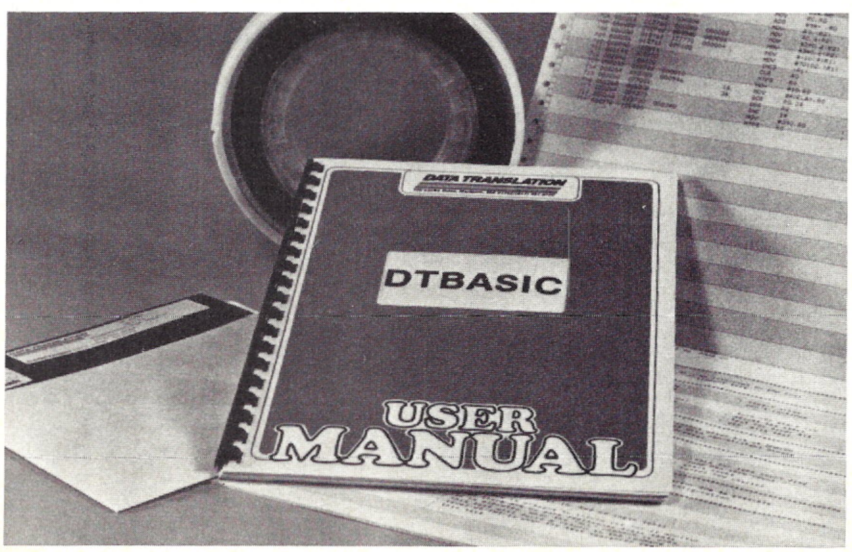
Third-party leasing is an attractive alternative to outright purchase of industrial robots, according to Rolec (Robotics Leasing Company) Ltd., a company that works with manufacturers who offer the leasing option as a part of their marketing. Leasing helps keep capital intact, improves cash flow, provides a hedge against inflation, and offers other benefits.

Now in its second year, Rolec offers true lease, finance lease, sale/leaseback, and other options. The company performs the credit investigation, creates a contract tailored to the customer's special needs, and sees that the robots are delivered and installed. Rolec then pays the manufacturer and begins collecting lease payments. They are exploring feasibility of a nationwide system of service centers. For more information, contact R. Courtenay Wallien, 3900 Chestnut St., Suite 324, Philadelphia, PA 19104. CIRCLE 23

Real-Time Data Acquisition Package Extends BASIC RT/11

DTBASIC is a conversational high-level programming language that permits high-performance data acquisition and control applications. The language combines BASIC/RT-11 with real-time subroutines for controlling a wide range of sophisticated data acquisition and control hardware. The real-time subroutines are written in MACRO-11 for speed. DTBASIC brings the ease of use of BASIC to a comprehensive selection of data acquisition hardware compatible with the LSI-11 bus. Analog input interfaces convert up to 64 channels of analog data with 12, 14, or 16 bits of resolution at sampling rates up to 135KHz. Output interfaces can provide 16 channels of analog data resolved to 12 bits. DTBASIC also supports up to 64 lines each of digital input and output.

DTBASIC is intended for use on DEC's LSI-11/2 and LSI-11/23 systems and Data Translation's LAB-DATAX. LSI-11 systems require at least 64K bytes of memory, the RT-11 operating system (version 4 or higher) and a programmable clock. Single quantity price is $1595 from Data Translation, 100 Locke Dr., Marlboro, MA 01752, (617) 481-3700. CIRCLE 24



Everything for LSI-11 Data Acquisition and Control

ADAC Corporation has released a new 48-page LSI-11 equipment catalog that provides quick-reference descriptions; features and photos of more than 40 A/D, D/A, and digital data conversion LSI-11 bus interface products; ADAC software offerings; and Prosys I measurement and control system. The product application tables help designers select both analog and digital LSI-11 interface equipment. A separate table provides application reference to analog I/O interfaces for non-LSI-11 bus applications. Each catalog has a companion price list and ordering guide. The catalog covers: mV level A/D conversion; temperature measurements; high-level, high-speed A/D conversions; TTL level digital I/O; pulse counters; optically isolated discrete I/O; contact closure sensing; discrete high-current outputs; programmable clock; serial interfaces; processor and memory boards; mass storage peripherals; operating systems; and high-level languages. The *Everything for LSI-11 Data Acquisition and Control* catalog and price list are available on request from Mr. Ben Minsk, ADAC Corporation, 70 Tower Office Park, Woburn, MA 01801, (617) 935-6668, Telex: 949329. CIRCLE 25

New Products

AIM/SYM Controller

UC/L System One is a combined hardware and software package that converts an AIM-65 or SYM-1 microcomputer into a programmable controller. The system does not require an editor, assembler, compiler or complex development system. Programs are entered on the microcomputer's keyboard eliminating the need for an external terminal. The UC/L programming language is designed for control applications. Operations include: input sensing, output switching, delay, stepping-motor control, and simple math functions. Programs can be linked with external code to provide functions not inherent to the system. Maximum program size is 170 lines with the minimum 1K bytes of programmable memory required by UC/L.

UC/L System One provides 16 channels of I/O sensing and control when used with the AIM-65. Fifteen channels are available with the SYM. The basic package including manual, UC/L software in 2716 EPROM, and an interface module and cable costs $175 from Polyarts Associates Inc., POB 21169, Seattle, WA 98111.
CIRCLE 26

Digital Image Display for Robotics Vision Applications

The Poynting Products Model 108 Digital Video Memory is a high-speed digital image display device that accommodates the medium-resolution display requirements of OEM, research, and maintenance environments. Organized as a 128 by 128 by 8-bit memory with continuous video output, the 108 is available in two versions: the 108-B is a functional 8.5- by 12-inch board requiring 1.8 Amps at 5V and 50 milliamps at - 12V for operation; the 108-RM consists of the 108-B together with internal power supply in a rack mountable chassis that features plug compatibility with the GE2210A Automation Camera Interface. The 108 is easily interfaced via parallel or direct memory access ports. Parallel input data is accepted at pixel rates of up to 3MHz over an 8-bit input data bus, allowing an image to be stored in as little as 5.4 ms. Only two input control signals are required: an End of Frame pulse indicates the start of a new image, and a Data Strobe pulse indicates the presence of the next image element on the input bus. Output video polarity is strap selectable, and personality modules allow the 108 to display images from a variety of cameras that provide 100 by 100 and 64 by 64 pixel images. Prices for the 108-B and 108-RM are $820 and $1120, respectively. For more information contact Poynting Products Inc., POB 1227, Oak Park, IL 60304, (312) 236-0725. CIRCLE 27

Visual Inspection System

The Inspector General, a computer-aided visual inspection system from Octek, performs at accuracy levels unattainable by human inspectors.

Octek specializes in the development of all types of inspection systems — optical character recognition (OCR), sorting, noncontact measurement, and defect detection. Available as component packages or turnkey systems, hardware, software, training, manuals, and other support necessary to improve productivity are always provided.

For more information, contact Chuck P. Comeau, Sales Manager, Octek, telephone (617) 273-0851. CIRCLE 28

72-RPM Synchronous Motor Specifications

Bodine Electric Company has released a catalog specifying the complete operating and application information on its new line of synchronous 72 rpm, 60Hz motors. The comprehensive HY-SYnc synchronous motor information can help reduce design time. Pertinent data supplied include torque-vs-inertia curves; maximum available torque; holding torque parameters; maximum available torque-vs-line voltage curves; plus general data, motor dimensions, and convenient formulas. For a copy of Catalog HY-1 write Bodine Electric Company, 2500 West Bradley Place, Chicago, IL 60618. CIRCLE 29

New Products

Intex-Talker

The Intex-Talker is a professional voice quality text-to-speech synthesizer that allows 64 programmable inflection levels. Available as either a stand-alone peripheral or basic board, the Intex-Talker provides a real-time audio interface for data processing, telecommunications, automation, education, and handicapped applications. A 6K-type text-to-phoneme translator provides accuracy unmatched by any other system. With RS-232C or parallel connectors using X-on/X-off handshaking, Intex-Talker can easily connect with all major computer and smart terminals. Intex-talker is priced at $295 and can be ordered from Intex Micro System Corporation, 4758 Brafferton, Bloomfield Hills, Michigan 48013, (313) 626-7454. CIRCLE 30

Apple Mimic Speech Processor

A Mimic Speech Processor has been developed for the Apple II+. Mimic is a low-cost multipurpose audio signal processing unit that converts an audio signal into a digital bit stream that can be sampled by computer, stored, and played back through the system. A data rate of 9600 bps produces speech quality and intelligibility acceptable for most applications. The MIMTALK BASIC software package simplifies vocabulary preparation. The Apple II+ compatible Mimic costs less than $200, and MIMTALK is available for $34.95. For more information contact Steve Thurston, Director of Marketing, Mimic Inc., POB 921, Acton, MA 01720, (617) 263-2101.
CIRCLE 31

New Sample-Hold Amplifier

The SHM-9 from Datel-Intersil is a self-contained sample-hold amplifier that combines high performance, versatility, and low cost. Internal circuitry includes a bipolar input amplifier, a low-leakage electronic switch, a FET output amplifier, a precision 1000pF hold capacitor, and logic control circuitry. The SHM-9 can be connected to virtually any A/D converter using the converters Start/Convert and Status signals. An internal inverter allows either a positive or negative Start/Convert command. Acquisition time for a 10V change to 0.01 percent is 6 microseconds and aperture delay time is 200 nseconds. The SHM-9 is suitable for numerous applications including sampling for A/D conversion, analog demultiplexing circuits, and simultaneous sampling circuits. Three models are available for operation over the commercial, industrial, and military temperature ranges. All models are packaged in a 16-pin, hermetically sealed, ceramic dual in-line package. Prices range from $39 to $89 depending on the temperature range and are available from Datel-Intersil, 11 Cabot Blvd., Mansfield, MA 02048, (617) 339-9341, Telex 951340.

CIRCLE 32

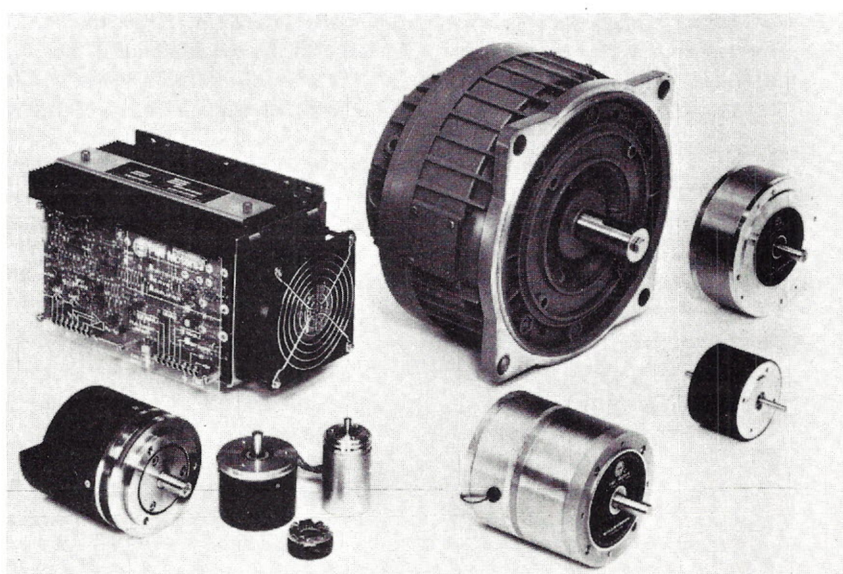
Motion-Control Components for Industrial Robots

PMI Motors has a complete line of motion-control components designed to produce the rapid start-stop capability and precise positioning accuracy required for industrial robots. The electric drive systems provide high repeatability factors.

High-performance DC servo-motors, using an ironless disk armature are available in fractional to integral horsepower sizes. The lack of rotating iron allows smooth torque at low speeds and rapid acceleration with high starting and peak torques.

In addition to motors, PMI offers precision gear reducers, optical encoders, analog tachometers, servo-amplifiers, and digital positioning controls. For further information contact Claude Chirignan, Chief Applications Engineer, PMI Motors Division, Kollmorgen Corporation, 5 Aerial Way, Syosset, NY 11791, (516) 938-8000.

CIRCLE 33



New Products

Opto-Isolated Microprocessor Controller

The Supper OSCCAM Control IV is an Intel 8085-A based controller that features three opto-isolated inputs and four opto-isolated outputs for operation that is free from line noise and RFI problems. Equipped with a silent electrosensitive printer and 7-digit LED display, the Supper OSCCAM IV is optionally available with an RS-232C serial interface. Priced at $3000, the Supper OSCCAM IV is available from Charles Supper Company Inc., 15 Tech Circle, Natick, MA 01760, (617) 655-4610.

CIRCLE 34

Digital Oscilloscope for Apple II

The Model 85 Digital Memory Oscilloscope from Northwest Instrument Systems Inc. plugs directly into two adjacent peripheral slots on an Apple II or II+ and provides two-channel waveform storage at a full 50 MHz band width. Waveforms are easily stored on disk for later use in calculations or displays. The Model 85 allows full program control of vertical attenuation, time base, and trigger level.

The provided machine-language system software occupies approximately 11.5K bytes of memory and uses both high-resolution graphics pages leaving 8K bytes available for a co-resident BASIC or machine-language program in a standard 48K-byte Apple II. Modular system software allows you to delete specific routines. A variety of control and demonstration software is available. For more information contact Northwest Instrument Systems Inc., POB 1309, Beaverton, OR 97075, (503) 297-1434.

CIRCLE 35

Turn a VIC-20 Into A Low Cost Development Computer With This Module

The Gloucester Computer Bus Co. Inc. PROMQUEEN CARTRIDGE provides EPROM programming, operating, and emulating capability for the Commodore VIC-20 computer. All necessary connections are made when the PROMQUEEN is plugged into the VIC's expansion port. The internal programming voltage power supply includes current limiting to prevent damage to the cartridge or the computer in the event of operator error or faulty EPROMS. The PROMQUEEN supports 2716, 2732A and 2732 EPROMs.

The PROMQUEEN contains 4 kilobytes of scratch memory for testing programs before burning them into EPROM. A MIMIC switch permits an external computer to access programs written into the PROMQUEEN scratch memory. In MIMIC mode a jumper cable connects between the PROMQUEEN zero insertion force (ZIF) socket and the socket in the external computer for which an EPROM is desired. This feature allows use of a VIC with PROMQUEEN as a development system. A DIP switch determines which of the four VIC expansion blocks is occupied by the PROMQUEEN. This allows the PROMQUEEN RAM to be used to expand the user BASIC memory of the VIC. It also allows the PROMQUEEN to be used in the Commodore Super Expander in conjunction with other cartridges, such as Commodore's VICMON, without address conflicts.

EPROMs in the zero insertion force socket can be run directly on the VIC. A socket isolate switch permits EPROM changes without removing VIC's power. The PROMQUEEN has a reset button for recovery from crashes, eliminating the need to store programs before testing them.

A toggle switch sets the PROMQUEEN for either 2716 or 2732A EPROMs, changing the voltage as required. 2732s can be programmed after readjusting a potentiometer.

A general purpose hex loader program on EPROM is included. The program loads into the VIC's memory, freeing the EPROM socket. The hex loader permits convenient loading of code for processors other than the VIC's 6502. It includes labels, block move, offset calculation, number conversion and more. It also runs the burn process, checks EPROM erasure, and verifies burns.

Software for storing BASIC programs on EPROM is included. The user's manual includes step by step instructions as well as tabular summaries of the Keystroke command options.

The PROMQUEEN CARTRIDGE costs $169.50, plus shipping. Contact Gloucester Computer Bus Co. Inc., 6 Brooks Road, Gloucester, MA 01930. Telephone (617) 283-7719.

CIRCLE 36

Classified Advertising

NEED HELP interfacing General Turtle Inc. controller Type CO11. Anyone out there have documentation? John Schaefer, 1487 College Ave., Palo Alto, CA 94306.

To $55K Openings nationwide for ME's and EE's in Robotics, Automation Machinery, controls, pneumatics, hydraulics, CNC, CAD/CAM, computers. Call or send resume to Victoria Tsitlik, Cross Country Consultants, 16 West 25th Street, Baltimore, MD. 21218. Tel. 301/889-2994. All Fees Paid.

POSITION OPEN: Mechanical Engineer wanted to help design mechanical components of mobile robot. Ground floor opportunity. Howard-Stephen Computers, Inc., 36 Summer St., Somerville, MA 02143 (617) 666-8080.

ROBOTICS INDUSTRY Directory Signed Collectors Edition — Only limited numbers of the 1981 ***Robotics Industry Directory***, the very first edition, are left. Originally thought to be sold out, a few boxes turned up in a garage (no kidding) — Phil Flora, Editor/Publisher, has signed them, adding to their authenticity and future value — But once they are gone, that's it! Available on a first-come, first-serve basis for $10 + $1.50 postage and handling from Robotics Industry Directory, POB 725, La Canada, CA 91011.

EXPERIMENTERS RESISTOR ASSORTMENT. 1/4W 5% carbon film. Common values like 330, 1K, 10K are 40 each. 10 each of less common values. 500 pieces $10. CI Electronics, Box 3034, Camarillo, CA 93011.

Robotics Classified Advertisement Order Form

Robotics Age classified advertisements reach thousands of people interested in designing and developing intelligent machines. To place a classified ad simply fill out the order form below and mail to *Robotics Age*, Strand Building, 174 Concord Street, Peterborough, N.H. 03458. The current word rate is $0.50 per word and can be paid by check, MasterCard, or VISA.

All classified advertisements are prepaid and accepted on a first come, first served basis for publication in the next available issue. Publishers reserve the right to reject any advertisement.

Please print:

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50

Word Count: ______ X $0.50 = Total Cost: ______

Payment:
MasterCard ☐ VISA ☐ Check ☐

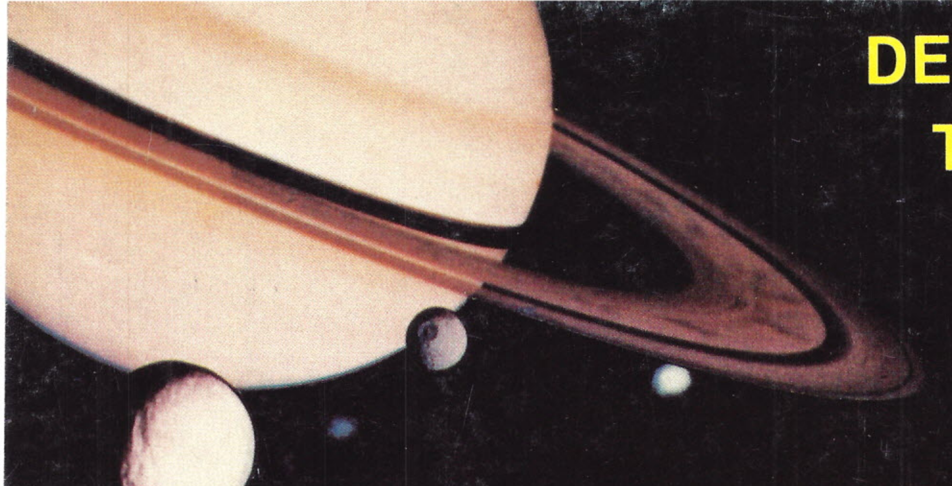
Name: ____________________
Company: ____________________
Street: ____________________
City/State: ______________ ZIP: ______

Signature: ____________________

Card Number: ☐☐☐☐☐☐☐☐☐☐☐☐

Interbank Number: ☐☐☐☐

Expiration Date: __/__



DESKTOP 68000 + TELESOFT[1] ADA[4] + PASCAL = *POWER*

COMPATIBILITY WITH DEC Q-BUS[3] AND STANDARD DEC[3] PERIPHERALS

THE IN/7000D BRINGS OUT OF THIS WORLD A/I PERFORMANCE WITHIN REACH

68000 BASED IN/7000D
Dramatically increased processing speed and flexibility:
- 8 MHz 68000 CPU
- 32-Bit internal arithmetic registers
- 24-Bit address register
- Powerful assembly language instructions support modular programming
- Designed for expansion to 32 bit word size
- 8 Levels of interrupt priority
- Vectored interrupts and DMA fully supported
- Outbenchmarks the IBM 370/145[2]*
- High speed string processing

Vastly increased memory:
- 68000 directly addresses 16 MB of memory
- Sorts can be done in core rather than with disk I/O

PASCAL AND TELESOFT[1] ADA[4]
- Fast program development
- Self documenting
- Supports structured programming
- Easy to update
- Easy to maintain
- Transportable from computer to computer
- Powerful logical constructs greatly simplify programming

For example: Modular procedures and functions
Strong data types
Case structures
- Local and global variables
- Recursive problem solving
- Block insert — a block of statements may be inserted anywhere one statement can exist
- Built in Boolean functions
- Library capability
- Program segmentability
- Procedure linking
- TeleSoft[1] Pascal and TeleSoft[1] Ada[4] translate to 68000 native code
- Built-in powerful string-handling features

TELESOFT[1] ADA[4]
- Designed to fulfill all DoD specifications
- INTRINSIC functions include:
  - * Multi-tasking and multi-programming
  - * Independent compilation of program units (called Packages)
- Fully implemented syntax checker which now parses the entire Ada[4] language

Saves development time.
- High speed sensory data processing
- High speed string processing power
- Fast coordinate transformations
- Easy implementation of in-memory AI algorithms, predicate calculus and trajectory computations
- Design and test algorithms quicker and easier
- Both Pascal and 68000 support features that make debugging far more efficient
- Plenty of memory, no need to use extra time for "programming tricks," previously needed with limited memory
- Mixed mode listing (Pascal source statements followed by 68000 statements)

UP TO 4 MByte OF RAM

*"Kilobaud Microcomputing" October, 1980
[1]A trademark of Renaissance Telesoftware Inc.
[2]A trademark of International Business Machines
[3]A trademark of Digital Equipment Corporation
[4]A trademark of Department of Defense

Call us now for more information, or fill in the coupon (301) 840-0700

I AM INTERESTED! Send more information now.

Name ______ Title ______
Company ______
Street ______
City ______ State ______ Zip ______
Phone ______ Call me IMMEDIATELY ☐
My need is: ☐ 1-3 mo. ☐ 3-6 mo. ☐ Future project ☐ File
☐ I am interested in a dealership.

INTELLIMAC/Intelligent Machines
6001 Montrose Road • Sixth Floor • Rockville, MD 20852
Phone: 301/984-8000
TWX: 710-828-9800 Cable: INTELLIMAC

INTELLIMAC
Intelligent Machines

6001 Montrose Road
Sixth Floor
Rockville, MD 20852
Phone: 301/984-8000
TWX: 710-828-9800
Cable: INTELLIMAC

Dealerships are available

CIRCLE 10

SATURN PHOTOGRAPH COURTESY OF NASA